



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1975

An alternative to nonlinear estimation in gun fire control systems

Brands, Robert Keith

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/20721>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



<http://www.nps.edu/library>

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

AN ALTERNATIVE TO NONLINEAR ESTIMATION
IN GUN FIRE CONTROL SYSTEMS

Robert Keith Brands

... LIBRARY
... POSTGRADUATE SCHOOL
... REY. CALIFORNIA 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

AN ALTERNATIVE TO NONLINEAR ESTIMATION
IN
GUN FIRE CONTROL SYSTEMS

by

Robert Keith Brands

December 1975

Thesis Advisor:

D. E. Kirk

Approved for public release; distribution unlimited.

T171660

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Alternative to Nonlinear Estimation in Gun Fire Control Systems		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1975
7. AUTHOR(s)		6. PERFORMING ORG. REPORT NUMBER
8. CONTRACT OR GRANT NUMBER(s)		
9. DONOR NAME AND ADDRESS State School California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. DOWNGRADING AND ADDRESS State School California 93940		12. REPORT DATE December 1975
13. E & ADDRESS (if different from Controlling Office) State School California 93940		13. NUMBER OF PAGES 149
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

EEF
Transactions on
Aerospace and
Electronic
Systems,
Vol 6, July 10.

rands

(of this Report)

public release; distribution unlimited.

(of the abstract entered in Block 20, if different from Report)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Kalman Filter Linear Estimator
Spherical Coordinate
Fire Control System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A Kalman filter with assumed linear system dynamics was derived from a spherical coordinate state vector and applied to a fire control system tracking environment. Two models of target dynamics were developed: a constant-velocity model and a correlated random acceleration model. The predicted position performance of the filters derived from the two basic models was optimized using Monte Carlo methods

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

against a set of test trajectories that include constant-velocity and maneuvering target profiles. The results of the Monte Carlo simulation of these filters are compared with the results obtained from a filter derived from a Cartesian Coordinate state vector. Switch-on-range adaptive filters were developed from the two basic models and evaluated by Monte Carlo methods. The results of the simulations of the adaptive filters with the two different state vectors are compared.

An Alternative to Nonlinear Estimation
in
Gun Fire Control Systems

by

Robert Keith Brands
Lieutenant, United States Navy
B.S., Moorhead State College, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

Thesis
B7988
C.1

ABSTRACT

A Kalman filter with assumed linear system dynamics was derived from a spherical coordinate state vector and applied to a fire control system tracking environment. Two models of target dynamics were developed: a constant-velocity model and a correlated random acceleration model. The predicted position performance of the filters derived from the two basic models was optimized using Monte Carlo methods against a set of test trajectories that include constant-velocity maneuvering target profiles. The results of the Monte Carlo simulation of these filters are compared with the results obtained from a filter derived from a Cartesian Coordinate state vector. Switch-on-range adaptive filters were developed from the two basic models and evaluated by Monte Carlo methods. The results of the simulations of the adaptive filters with the two different state vectors are compared.

TABLE OF CONTENTS

I.	INTRODUCTION - - - - -	12
A.	AN ALTERNATIVE TO NONLINEAR ESTIMATION IN GUN FIRE CONTROL SYSTEMS - - - - -	12
B.	LINEAR FILTERING THEORY- - - - -	17
II.	DEVELOPMENT OF THE FILTER MODELS - - - - -	20
A.	CONSTANT-VELOCITY ($1/S^2$) MODEL - - - - -	20
B.	CORRELATED RANDOM ACCELERATION MODEL - - - - -	23
III.	RESULTS OF THE MONTE CARLO SIMULATION OF THE PROPOSED LINEAR FILTERS- - - - -	28
A.	EXPERIMENTAL PROCEDURE - - - - -	28
B.	SIMULATION RESULTS FOR THE CONSTANT-VELOCITY MODEL- - - - -	31
C.	SIMULATION RESULTS FOR THE CORRELATED RANDOM ACCELERATION MODEL - - - - -	37
IV.	ADAPTIVE FILTERING - - - - -	48
A.	THE ADAPTIVE BANDWIDTH FILTER- - - - -	48
B.	SIMULATION RESULTS FOR THE CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTER- - - - -	50
C.	SIMULATION RESULTS FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER- - - - -	50
V.	SUMMARY AND CONCLUSIONS- - - - -	60
A.	SUMMARY OF THE TEST RESULTS- - - - -	60
1.	Summary of the Simulation Results for the Two Models - - - - -	60
2.	Summary of Simulation Results for the Adaptive Filters - - - - -	61
B.	CONCLUSIONS- - - - -	61
C.	SUGGESTIONS FOR FURTHER INVESTIGATION- - - - -	62

APPENDIX A:	MONTE CARLO SIMULATION PROGRAM-	- - - - -	64
A.1	PROGRAM LISTING AND DESCRIPTION	- - - - -	64
A.2	LISTING OF KEY VARIABLES USED IN THE MONTE CARLO SIMULATION-	- - - - -	-112
A.3	IBM SYSTEM 360/67 SOURCE LIBRARY SUBROUTINES CALLED BY THE MONTE CARLO SIMULATION PROGRAM-	- - - - -	-115
A.4	MONTE CARLO SIMULATION PROGRAM FLOWCHART-		-116
APPENDIX B:	TEST TRAJECTORIES	- - - - -	-131
B.1	DESCRIPTION OF TEST TRAJECTORY PROFILES	- - - - -	-131
B.2	TRAJECTORY GENERATING PROGRAM	- - - - -	-135
APPENDIX C:	THE N-STEP PREDICTION PROBLEM	- - - - -	-136
BIBLIOGRAPHY	- - - - -	- - - - -	-148
INITIAL DISTRIBUTION LIST-	- - - - -	- - - - -	-149

LIST OF TABLES

Table		Page
I	DIAGONAL ELEMENTS OF THE OPTIMIZED $\tilde{Q}(K)$ MATRICES FOR THE CONSTANT-VELOCITY MODEL FILTERS- - - - -	32
II	SIMULATION RESULTS FOR THE OPTIMIZED CONSTANT-VELOCITY MODEL FILTERS- - - - -	34
III	COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN COORDINATE STATE VECTOR CONSTANT-VELOCITY MODEL FILTERS- - - - -	36
IV	POSITION ESTIMATION PERFORMANCE OF THE OPTIMIZED CONSTANT-VELOCITY MODEL FILTERS- - -	38
V	RATE ESTIMATION PERFORMANCE OF THE OPTIMIZED CONSTANT-VELOCITY MODEL FILTERS- - -	39
VI	DIAGONAL ELEMENTS OF THE OPTIMIZED $\tilde{Q}(K)$ MATRICES FOR THE CORRELATED RANDOM ACCELERATION MODEL FILTERS - - - - -	40
VII	SIMULATION RESULTS, FOR THE CORRELATED RANDOM ACCELERATION MODEL FILTERS- - - - -	43
VIII	COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN COORDINATE STATE VECTOR CORRELATED RANDOM ACCELERATION MODEL FILTERS- - - - -	44
IX	POSITION ESTIMATION PERFORMANCE OF THE OPTIMIZED CORRELATED RANDOM ACCELERATION MODEL FILTERS- - - - -	45
X	RATE ESTIMATION PERFORMANCE OF THE OPTIMIZED CORRELATED RANDOM ACCELERATION MODEL FILTERS- - - - -	46
XI	ACCELERATION ESTIMATION PERFORMANCE OF THE OPTIMIZED CORRELATED RANDOM ACCELERATION MODEL FILTERS- - - - -	47
XII	SWITCH-ON-RANGE ADAPTIVE FILTER SIMULATION FOR THE CONSTANT-VELOCITY MODEL FILTER - - - - -	51

XIII	COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN COORDINATE STATE VECTOR CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTERS - - - - -	52
XIV	POSITION ESTIMATION PERFORMANCE FOR THE CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTER- - - - -	53
XV	RATE ESTIMATION PERFORMANCE FOR THE CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTER- - - - -	54
XVI	SWITCH-ON-RANGE ADAPTIVE FILTER SIMULATION FOR THE CORRELATED RANDOM ACCELERATION MODEL- - - - -	56
XVII	POSITION ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER- - - - -	57
XVIII	RATE ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH- ON-RANGE ADAPTIVE FILTER - - - - -	58
XIX	ACCELERATION ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH- ON-RANGE ADAPTIVE FILTER - - - - -	59

LIST OF FIGURES

Figure		Page
1	FIRE CONTROL SYSTEM SIMPLIFIED BLOCK DIAGRAM- - - - -	13
2	THE SPHERICAL COORDINATE SYSTEM- - - - -	15
3	CONSTANT-VELOCITY PLANT DRIVEN BY CORRELATED NOISE - - - - -	25
4	GEOMETRY OF THE PREDICTED POSITION ERROR PROBLEM IN A SPHERICAL COORDINATE SYSTEM - - -	142
5	THE POLAR PLANE AFTER ROTATION - - - - -	143
6	FIXED RANGE PROJECTION ONTO THE POLAR PLANE AFTER ROTATION - - - - -	146

TABLE OF SYMBOLS AND ABBREVIATIONS

$\tilde{x}(K)$	n-dimensional state vector at time $t = KT$.
$\tilde{w}(K)$	p-dimensional random forcing vector at time $t = KT$.
$\tilde{\phi}(K+1,K)$	(n x n)-dimensional state transition matrix.
$\tilde{\Gamma}(K)$	(n x p)-dimensional random forcing distribution matrix.
$\tilde{z}(K)$	m-dimensional vector of observations at time $t = KT$.
$\tilde{H}(K)$	(m x n)-dimensional observation matrix.
$\tilde{v}(K)$	m-dimensional measurement noise vector at time $t = KT$.
T	sampling period.
K	a non-negative integer denoting the current sample.
E	expectation operator.
δ_{kj}	Kronecker Delta Function.
$\tilde{Q}(K)$	(p x p)-dimensional covariance of random forcing matrix at time $t = KT$.
$\tilde{R}(K)$	(m x m)-dimensional covariance of measurement noise at time $t = KT$.
$\hat{\tilde{x}}(K/K)$	n-dimensional vector of state estimates at time $t = KT$, given measurements up to and including time $t = KT$.
$\hat{\tilde{x}}(K/K-1)$	n-dimensional vector of the one-step state prediction.
$\tilde{G}(K)$	(n x m)-dimensional Kalman Gain matrix at time $t = KT$.
$\tilde{P}(K/K)$	(n x n)-dimensional covariance of estimation error matrix.
$\tilde{P}(K+1/K)$	(n x n)-dimensional one-step prediction error covariance matrix.

I (n x n)-dimensional identity matrix.
 $[\]^T$ matrix(vector) transpose.
 $[\]^{-1}$ matrix inverse.

I. INTRODUCTION

A. AN ALTERNATIVE TO NONLINEAR ESTIMATION IN GUN FIRE CONTROL SYSTEMS

The increased performance characteristics of modern aircraft and missiles has made apparent the need for more effective fire control systems. With the introduction of digital fire control computers, Kalman filtering techniques offer an attractive means of improving the effectiveness of a fire control system. Indeed, several recent studies conducted at the Naval Postgraduate School [Refs. 1-6] have specifically addressed the application of Kalman filtering techniques to fire control systems. It is interesting to note that the references cited above all have one distinctive feature in common; they use Cartesian Coordinate state vectors. A Cartesian Coordinate state vector offers the attraction of linear plant dynamics for a constant-velocity target but has the disadvantage of nonlinear measurement equations when used in a fire control system that measures the target's position in spherical coordinates. One possible solution to the nonlinear measurement problem is to process the sensor measurements through a prefilter as in Figure 1. The output of the sensor consists of noise corrupted measurements of the target's position in spherical coordinates

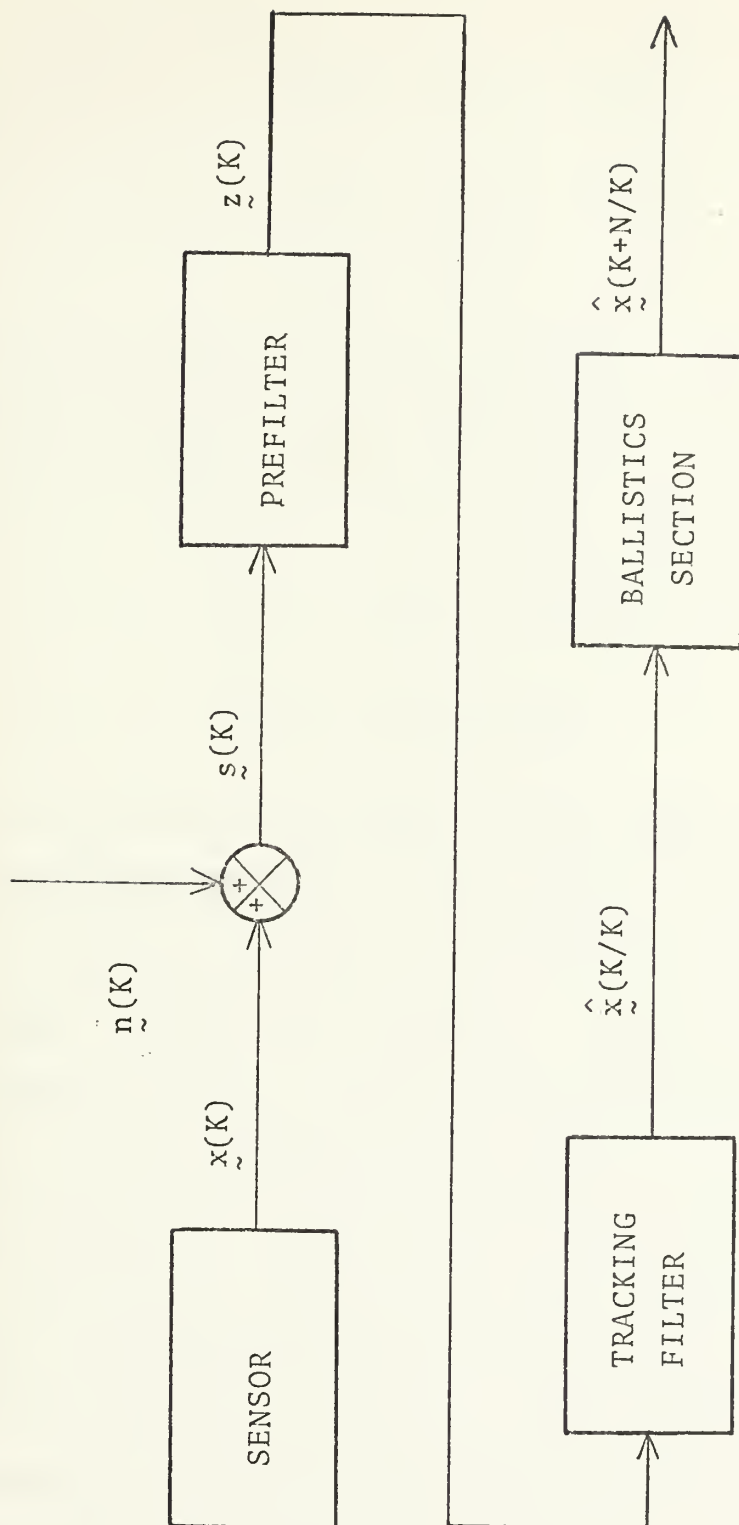


Fig. 1. FIRE CONTROL SYSTEM SIMPLIFIED BLOCK DIAGRAM

$$\begin{aligned}
s_1(K) &= r(K) + n_1(K) \\
s_2(K) &= \phi(K) + n_2(K) \\
s_3(K) &= \theta(K) + n_3(K)
\end{aligned}
\tag{1.1}$$

which serve as inputs to the prefilter. The appropriate coordinate transformation equations from Figure 2 are

$$\begin{aligned}
x(K) &= r(K)\sin\phi(K)\cos\theta(K) \\
y(K) &= r(K)\cos\phi(K)\cos\theta(K) \\
z(K) &= r(K)\sin\theta(K)
\end{aligned}
\tag{1.2}$$

where it has been shown [Ref. 3] that $\tilde{y}(K)$, the measurement noise process presented to the tracking filter is a non-linear, position-dependent function of the variances of the range, bearing, and elevation errors. Clearly then, the measurements become correlated in the coordinate conversion process. Full implementation of such a filtering scheme could conceivably involve much on-line computation since the statistics for $\tilde{y}(K)$ are no longer stationary. Another distinct disadvantage of the prefiltering method lies in the errors generated due to the added finite precision arithmetic of the coordinate conversion process. Such errors may be qualitatively described as a noise process whose statistical properties are difficult at best to identify. An excellent treatment of this specific topic is found in Gelb [Ref. 7].

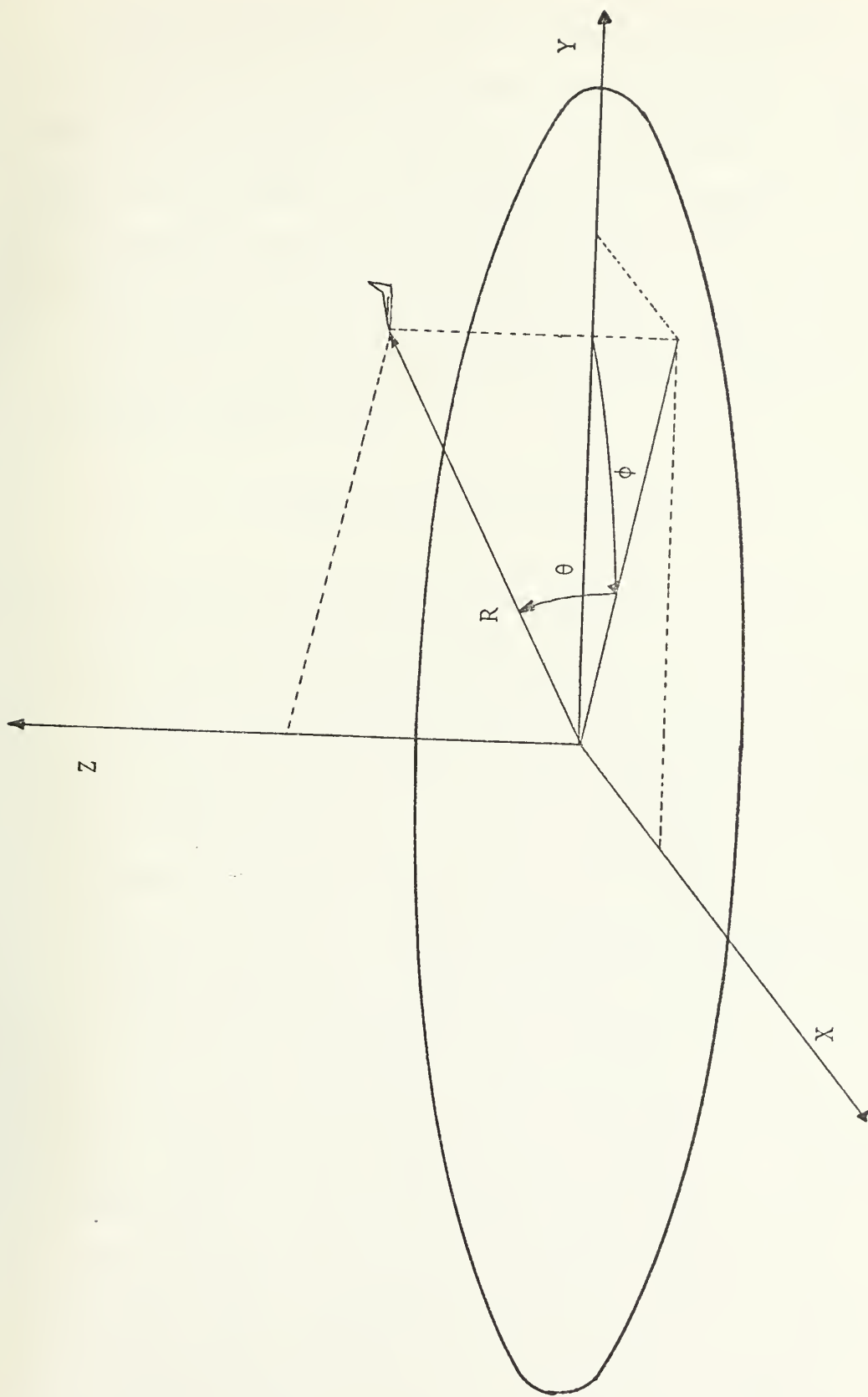


Fig. 2. THE SPHERICAL COORDINATE SYSTEM

Another approach to the nonlinear measurement process is to linearize the measurement equations about some reference or nominal trajectory. Unfortunately, such an approach is usually accomplished by the implementation of an extended or iterative extended Kalman filter with the attendant increase in on-line computing requirements.

Consider the filtering problem with a spherical coordinate state vector

$$x(K) = [r, \dot{r}, \ddot{r}, \phi, \dot{\phi}, \ddot{\phi}, \theta, \dot{\theta}, \ddot{\theta}]^T \quad (1.4)$$

Such a state vector has the advantage of linear measurement equations eliminating the correlation and position dependence of the measurement noise process. The disadvantage of a spherical state vector lies in the coupled and nonlinear system-target-dynamics, even for a constant-velocity target. Once again, the implementation of this filter would normally be with an extended or iterative extended Kalman filter. This thesis proposes a filter having a spherical coordinate state vector with assumed linear system dynamics. In view of the fact that any system model is an approximation to a physical system, this seems to be a reasonable assumption. Implementation of this would be a great deal simpler than the previously mentioned filters. A filter designed on this basis would then of necessity be sub-optimal. However, given acceptable performance, the sub-optimal filter might be preferred to the optimal nonlinear filter. Faced then,

with a sub-optimal filter due to modeling inaccuracies, a technique must be found to compensate for the unmodeled nonlinear properties of the system. Jazwinski [Ref. 8] has proposed two methods for the compensation of inaccurate a priori knowledge of the system. The first involves a fictitious noise input to the system, while the second overweights the most recent data. The net result of either method is an increase in the filter bandwidth, making the filter more responsive to system transients but increasing its sensitivity to the measurement noise process. The fictitious noise input method was chosen as the most promising and was used in this thesis. So that some quantifiable measure of the feasibility of these filters is possible, this thesis paralleled Parr's work [Ref. 4] and comparisons are made with his results.

B. LINEAR FILTERING THEORY

The system studied in this thesis is composed of a target and a sensor which gives noise-corrupted measurements of the target's position in spherical coordinates. The system may be approximated by the linear difference equations

$$\begin{aligned}\tilde{x}(K+1) &= \tilde{\phi}(K+1,K)\tilde{x}(K) + \tilde{\Gamma}(K)\tilde{w}(K) \\ \tilde{z}(K) &= \tilde{H}(K)\tilde{x}(K) + \tilde{v}(K)\end{aligned}\tag{1.5}$$

The statistics assumed for the stochastic processes $\tilde{w}(K)$

and $\underline{v}(K)$ are

$$E[\underline{w}(K)] = E[\underline{v}(K)] = \underline{0} \quad \text{for all } K \quad (1.6)$$

$$E[\underline{v}(k)\underline{v}^T(j)] = \underline{R}(K)\delta_{kj} \quad (1.7)$$

$$E[\underline{w}(k)\underline{w}^T(j)] = \underline{Q}(K)\delta_{kj} \quad (1.8)$$

$$E[\underline{w}(K)\underline{v}^T(K)] = E[\underline{v}(K)\underline{w}^T(K)] = \underline{0} \quad \text{for all } K \quad (1.9)$$

$$E[\underline{w}(K)\underline{x}^T(0)] = E[\underline{x}(0)\underline{w}^T(K)] = \underline{0} \quad \text{for all } K \quad (1.10)$$

$$E[\underline{v}(K)\underline{x}^T(0)] = E[\underline{x}(0)\underline{v}^T(K)] = \underline{0} \quad \text{for all } K \quad (1.11)$$

The linear, discrete Kalman filter is characterized by the difference equations

$$\hat{\underline{x}}(K/K) = \hat{\underline{x}}(K/K-1) + \underline{G}(K) [\underline{z}(K) - \underline{H}(K)\hat{\underline{x}}(K/K-1)] \quad (1.12)$$

with the one-step prediction given by

$$\hat{\underline{x}}(K/K-1) = \underline{\phi}(K, K-1)\hat{\underline{x}}(K-1/K-1) \quad (1.13)$$

The Kalman Gain matrix, $\underline{G}(K)$, is found by solving the following equations

$$\underline{G}(K) = \underline{P}(K/K-1)\underline{H}^T(K) [\underline{H}(K)\underline{P}(K/K-1)\underline{H}^T(K) + \underline{R}(K)]^{-1} \quad (1.14)$$

$$\underline{P}(K/K) = [\underline{I} - \underline{G}(K)\underline{H}(K)]\underline{P}(K/K-1) \quad (1.15)$$

$$\begin{aligned} \underline{P}(K+1/K) = & \underline{\phi}(K+1, K)\underline{P}(K/K)\underline{\phi}^T(K+1, K) + \\ & \underline{\Gamma}(K)\underline{Q}(K)\underline{\Gamma}^T(K) \end{aligned} \quad (1.16)$$

To start the gain calculations, the matrix $\underline{P}(K/K-1)$ must

be initialized. Taking the standard approach, $\tilde{P}(0/-1)$ is given by

$$\tilde{P}(0/-1) = E[(\tilde{x}(0) - \bar{\tilde{x}}_0)(\tilde{x}(0) - \bar{\tilde{x}}_0)^T] \quad (1.17)$$

where

$$E[\tilde{x}(0)] = \bar{\tilde{x}}_0 \quad (1.18)$$

An unbiased estimator is obtained if the filter is started by setting

$$\hat{\tilde{x}}(0/-1) = \bar{\tilde{x}}_0 \quad (1.19)$$

II. DEVELOPMENT OF THE FILTER MODELS

A. CONSTANT-VELOCITY ($1/S^2$) MODEL

The first filter tested was based on a dynamic system model that assumes linear constant-velocity target motion in each of three coordinate directions.

The state transition matrix $\phi(K+1,K)$ and the random forcing input distribution matrix $\Gamma(K)$ are

$$\phi(K+1,K) = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

and

$$\Gamma(K) = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & \frac{T^2}{2} \\ 0 & 0 & T \end{bmatrix} \quad (2.2)$$

where T is the sampling period.

Since the state vector is expressed in spherical coordinates, the measurement process is linear and is given by

$$\tilde{z}(K) = \tilde{H}(K)\tilde{x}(K) + \tilde{v}(K) \quad (2.3)$$

The measurement noise processes for the individual coordinate directions are assumed to be independent processes. The covariance of measurement noise matrix $\tilde{R}(K)$ is diagonal and is given by

$$\tilde{R}(K) = \begin{bmatrix} \sigma_R^2 & & 0 \\ & \sigma_B^2 & \\ 0 & & \sigma_E^2 \end{bmatrix} \quad (2.4)$$

Because of its simplicity the constant-velocity model cannot realistically be expected to perform well against all types of target trajectories and may not perform acceptably against maneuvering targets. The performance of the constant-velocity model filter can be improved by the use of the system random input covariance matrix $\tilde{Q}(K)$. Indeed, one of the functions of the $\tilde{Q}(K)$ matrix is to compensate for the inadequacies of a simplified model, and for this specific thesis, $\tilde{Q}(K)$ can also partially account for the system nonlinearities not explicitly included in the system model. The subject of the sensitivity of the

filter's N-step prediction performance to the $Q(K)$ matrix forms the basis for the simulations of Chapter III.

Initialization of the filter requires a priori knowledge of the statistics of the system initial conditions so that the gain calculations may be started. For any fixed coordinate system, it is reasonable to consider the mean of all possible system initial conditions as zero or

$$E[\underline{\tilde{x}}(0)] = \underline{\bar{x}}_0 = \underline{0} \quad (2.5)$$

It has been shown [Refs. 7 and 9] that under proper initialization, the filter will remain unbiased for $K \geq 0$, i.e.,

$$E[\underline{\tilde{e}}(K/K)] = \underline{0} \quad \text{for all } K \quad (2.6)$$

where

$$\underline{\tilde{e}}(K/K) = \underline{\hat{x}}(K/K) - \underline{\tilde{x}}(K) \quad (2.7)$$

The unbiased condition is satisfied if

$$\underline{\hat{x}}(0/-1) = \underline{\bar{x}}_0 \quad (2.8)$$

Initialization of the covariance of estimation error matrix requires a somewhat different approach. Following Parr [Ref. 4] it is quantitatively shown that if the diagonal elements of the initial covariance of estimation error matrix are made very large, the net effect is that the first position estimate is made identically equal to

the first system measurement. In this manner the system is forced into tracking the target rapidly. For all simulations conducted then, it was assumed that

$$\tilde{P}(0/-1) = \begin{bmatrix} 10^8 & 0 \\ 0 & 10^8 \end{bmatrix} \quad (2.9)$$

B. CORRELATED RANDOM ACCELERATION MODEL

The constant-velocity model discussed in the previous section views any target maneuvers as small perturbations about a nominal, constant-velocity trajectory. This tacitly assumes that any target maneuvers are uncorrelated in time, or from one measurement to the next. Target maneuvers will generally take place over a period of n measurements and the magnitude of n is dependent on the level of the target's maneuver, i.e., for a slow turn n is relatively large whereas for an evasive type maneuver n is smaller. The fact that the accelerations occur over a finite period of time correlates the target's accelerations (maneuvers) in time. One approach to modeling the effects of time-correlated target maneuvers is to use a constant-velocity plant driven by correlated noise obtained from a low-pass filter of the form

$$T(s) = \frac{\alpha}{(s + \alpha)} \quad (2.10)$$

driven by white noise. The constant α is interpreted as the time constant or degree of time correlation of the target's maneuver.

For one dimension, the overall model transfer function is

$$T(s) = \frac{\alpha}{s^2(s + \alpha)} \quad (2.11)$$

as depicted in Figure 3.

For the three-dimensional case, the filter is ninth order and has the state vector

$$\tilde{x}(K) = [r, \dot{r}, \ddot{r}, \phi, \dot{\phi}, \ddot{\phi}, \theta, \dot{\theta}, \ddot{\theta}]^T \quad (2.12)$$

The correlated random acceleration model is discussed in detail by McKinley [Ref. 5] and Singer [Ref. 10]. From the cited references, the (3 x 3) state transition matrix

$\phi_i(T)$ and the (3 x 1) random forcing input vector $\tilde{\Gamma}_i(T)$ are for the single dimension with assumed linear system dynamics

$$\phi_i(T) = \begin{bmatrix} 1.0 & T & \frac{T}{\alpha} - \frac{1}{\alpha^2} (1 - e^{-\alpha T}) \\ 0.0 & 1.0 & \frac{1}{\alpha} (1 - e^{-\alpha T}) \\ 0.0 & 0.0 & e^{-\alpha T} \end{bmatrix} \quad (2.13)$$

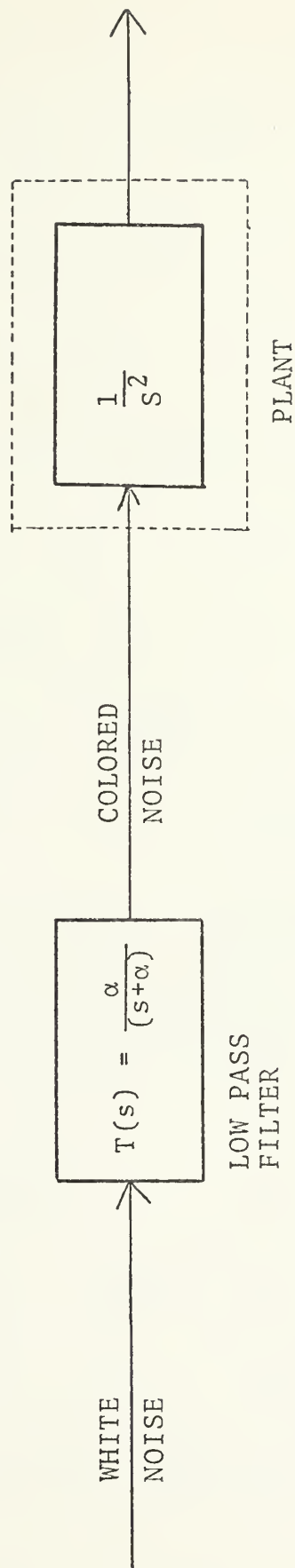


Fig. 3. CONSTANT-VELOCITY PLANT DRIVEN BY CORRELATED NOISE

and

$$\tilde{\Gamma}_i(T) = \begin{bmatrix} \frac{T^2}{2} - \frac{T}{\alpha} + \frac{1}{\alpha^2} (1 - e^{-\alpha T}) \\ T - \frac{1}{\alpha} (1 - e^{-\alpha T}) \\ (1 - e^{-\alpha T}) \end{bmatrix} \quad (2.14)$$

For the ninth-order filter the state transition matrix $\phi(K+1, K)$ is formed by

$$\phi(K+1, K) = \begin{bmatrix} \phi_r(T) & 0 & 0 \\ 0 & \phi_B(T) & 0 \\ 0 & 0 & \phi_E(T) \end{bmatrix} \quad (2.15)$$

where $\phi_r(T)$, $\phi_B(T)$, and $\phi_E(T)$ are individually given by $\phi_i(T)$ in Equation (2.13). In a similar manner, for the ninth-order filter the three-dimensional system matrix $\tilde{\Gamma}(K)$ is given by

$$\tilde{\Gamma}(K) = \begin{bmatrix} \tilde{\Gamma}_r(T) & 0 & 0 \\ 0 & \tilde{\Gamma}_B(T) & 0 \\ 0 & 0 & \tilde{\Gamma}_E(T) \end{bmatrix} \quad (2.16)$$

where $\tilde{\Gamma}_r(T)$, $\tilde{\Gamma}_B(T)$, $\tilde{\Gamma}_E(T)$ are individually given by $\tilde{\Gamma}_i(T)$ in Equation (2.14).

The covariance of measurement noise matrix $\tilde{R}(K)$ is the same as given in the preceding section by Equation (2.4). Initialization of the filter is carried out in exactly the same fashion as described in the previous section by Equations (2.5), (2.8), and (2.9). As in the preceding section, $\tilde{Q}(K)$, the covariance of the random forcing input white noise process presented to the coloring filter, was used to compensate for the unmodeled dynamic nonlinearities. The sensitivity of the filter's N-step prediction performance to the $\tilde{Q}(K)$ matrix selected is treated in Chapter III.

In the interest of saving computer time, the correlated random acceleration model was tested using a single value for α , the maneuver time constant. Since Parr [Ref. 4] reported that the best overall performance in his simulations was obtained from a value of $\alpha = 0.5$, that value was used in the study of this model. Additionally, the maneuver time constant α , was fixed at the same value for each of the three coordinate directions, though this constraint need not necessarily be imposed on the system.

III. RESULTS OF THE MONTE CARLO SIMULATION OF THE PROPOSED LINEAR FILTERS

A. EXPERIMENTAL PROCEDURE

Monte Carlo simulation techniques were used in the optimization and testing of the filters developed in Chapter II. A listing and description of the FORTRAN program used is given in Appendix A. All results are the products of 100-member ensemble simulations. Test trajectories 1-8 were used in the optimization and testing process and Appendix B gives a description of the profile for each trajectory. The profiles of trajectories 1, 2, 3, and 4 are identical to the short-range constant-velocity, short-range maneuvering, long-range constant-velocity, and long-range maneuvering profiles, respectively, used by Parr [Ref. 4]. Trajectories 5-8 were included to examine the filter's performance against more severe target maneuvers.

The elements of $\tilde{R}(K)$, the covariance of measurement noise process matrix, were fixed for all simulations at

$$\begin{aligned}\sigma_R &= 3.0 \text{ yds.} \\ \sigma_B &= 0.01722 \text{ Deg.} \\ \sigma_E &= 0.01722 \text{ Deg.}\end{aligned}\tag{3.1}$$

The values given in Equation (3.1) are those used by Parr [Ref. 4] and are used in this thesis in an attempt to

duplicate Parr's testing environment, and minimize procedural discrepancies.

The performance evaluation of the filters developed in Chapter II is based on their N-step prediction or position-state prediction accuracy. A detailed description of the method used in the determination of the error in the predicted position is given in Appendix C. The results presented in this chapter are the percentages of "hits" and "scares" computed over the entire ensemble. A "hit" is defined as an error in the predicted position of less than eight yards, while a "scare" is defined as an error in the predicted position of less than 16 yards, but greater than eight yards.

Initialization of each filter simulated was identical in all trials and is described in Chapter II by Equations (2.5), (2.8), and (2.9).

The elements of the covariance of the random forcing input matrix $\tilde{Q}(K)$ were determined by a two-dimensional optimization search. The assumption of linear uncoupled dynamics eliminates any spatial correlation, resulting in a diagonal $\tilde{Q}(K)$ matrix of the form

$$\tilde{Q}(K) = \begin{bmatrix} q_r & 0 & 0 \\ 0 & q_B & 0 \\ 0 & 0 & q_E \end{bmatrix} \quad (3.2)$$

If the system dynamics were truly linear, then the diagonal elements of (3.2) would represent the variances of the random forcing input to the system. However, since $\underline{Q}(K)$ is also used to compensate for inaccuracies in the system model, q_r , q_B and q_E must be adjusted accordingly. To simplify the optimization process for both models, q_B and q_E were constrained to be equal throughout. The two-dimensional search optimized the filter's position-state prediction accuracy as a function of two variables, q_r and the pair $q_B = q_E$.

In the case of the constant-velocity model, the elements of the matrix $\underline{Q}(K)$ serve as a means by which the bandwidth of the filter is increased to ensure that the filter will respond to target accelerations. This, of course, is in addition to compensation for the unmodeled nonlinearities of the system dynamics. Because of the dual role played by the matrix $\underline{Q}(K)$, a brute force search was done to determine the elements of $\underline{Q}(K)$ that gave the best position-state prediction performance. In a similar manner, for the filter based on the correlated random acceleration model, the bandwidth must be increased to permit sufficient response to targets exhibiting non-zero acceleration rates. Here again brute force search techniques were used to determine the $\underline{Q}(K)$ matrix that gave the best position prediction performance.

For both models, once the optimum $\underline{Q}(K)$ matrix was identified for a specific trajectory, that particular

filter configuration was tested against the remaining trajectories. The net result being that each optimized filter was tested against seven additional trajectories to ensure that the optimum $\tilde{Q}(K)$ had been found, and to determine the effect of using a $\tilde{Q}(K)$ selected for a track of one type against targets having different maneuvering characteristics.

B. SIMULATION RESULTS FOR THE CONSTANT-VELOCITY (L/S^2) MODEL

The results presented in this section are for the optimized filter configurations. The term "Filter 1" denotes the filter with the specific $\tilde{Q}(K)$ matrix that gave the best position prediction performance against trajectory 1. Thus, the simulation results for eight filters, or more precisely, eight filter configurations are presented.

As pointed out in the preceding section, the assumption of linear uncoupled dynamics results in the diagonal $\tilde{Q}(K)$ matrix given by Equation (3.2). Table I gives the diagonal elements of the $\tilde{Q}(K)$ matrix that provided the best position prediction performance for each test trajectory. The relative magnitude of the diagonal elements for the optimal filters are as expected. That is to say, the elements of $\tilde{Q}(K)$ for the non-maneuvering trajectories (1, 3, 7, 8) are lower in magnitude than those for the maneuvering trajectories (2, 4, 5, 6). Similarly, the elements of $\tilde{Q}(K)$ for the long-range maneuvering trajectories are lower in magnitude than those for the short-range maneuvering trajectories. Similar

Table I
 DIAGONAL ELEMENTS OF THE OPTIMAL $\tilde{Q}(K)$ MATRICES
 FOR THE CONSTANT-VELOCITY ($1/S^2$) MODEL

FILTER	q_{11}	q_{22}	q_{33}
1	1.00×10^{-6}	8.58×10^{-9}	8.58×10^{-9}
2	0.05	0.1153	0.1153
3	1.00×10^{-6}	1.13×10^{-10}	1.13×10^{-10}
4	0.025	0.006553	0.006553
5	0.0075	0.1372	0.1372
6	1.00×10^{-4}	5.55×10^{-7}	5.55×10^{-7}
7	0.01	3.37×10^{-7}	3.37×10^{-7}
8	0.005	1.66×10^{-7}	1.66×10^{-7}

observations may be made for the moderate level maneuvers as opposed to the evasive level maneuvers. Note that the elements of $\underline{Q}(K)$ for the long-range evasive maneuver profile (trajectory 6) are smaller than those for the long-range moderate level maneuver profile (trajectory 4). This appears to be a reasonable phenomenon in light of the fact that overweighting the estimates for a faster more highly maneuvering target at long range could result in larger errors in the predicted position. It follows then that the bandwidth of the optimal filter for the long-range evasive maneuver should be somewhat smaller than the bandwidth for the moderate level maneuver at long range.

Table II gives the simulation results for the optimal filter configurations as well as for each optimal filter configuration against the other trajectories. The results shown are not surprising in that for a specific trajectory, the optimal filter configuration gave the best results. In a few isolated cases, other filter configurations gave results that approach those for the optimal configuration. Though this was not expected, the differences are not considered small enough to justify any additional effort in the optimization process.

Table III compares the hit/scare percentages from the simulations of the two models using the two different state vectors. The results for the spherical coordinate state vector represent the optimal performance against a specific

Table II

SIMULATION RESULTS FOR THE OPTIMIZED
CONSTANT-VELOCITY MODEL FILTERS

FILTER	Trajectory 1		Trajectory 2		Trajectory 3		Trajectory 4	
	%hit	%scare	%hit	%scare	%hit	%scare	%hit	%scare
1	86.89	5.00	9.02	11.45	40.21	22.29	0.0	0.10
2	71.77	16.48	25.34	24.05	0.71	3.62	0.1	0.86
3	86.86	4.91	9.02	11.45	40.81	21.14	0.0	0.10
4	85.82	5.84	20.57	26.11	5.10	19.10	0.71	0.95
5	70.34	17.27	24.84	24.66	0.0	4.76	0.0	0.0
6	85.23	4.68	9.02	11.45	40.03	21.32	0.0	0.0
7	86.59	4.91	9.02	11.45	40.24	22.43	0.0	0.0
8	86.62	4.93	9.02	11.45	40.38	22.29	0.0	0.0

Table II

SIMULATION RESULTS FOR THE OPTIMIZED

CONSTANT-VELOCITY MODEL FILTERS (Cont'd)

FILTER	Trajectory 5		Trajectory 6		Trajectory 7		Trajectory 8	
	%hit	%score	%hit	%score	%hit	%score	%hit	%score
1	12.17	16.05	7.00	0.14	85.22	5.35	69.10	11.36
2	27.62	11.64	0.46	1.57	61.80	20.17	3.57	14.29
3	12.43	15.79	7.00	0.14	86.07	5.35	69.09	11.36
4	28.95	12.40	0.04	0.11	82.70	8.62	27.29	37.25
5	29.33	12.17	0.0	0.0	59.87	20.72	4.25	12.71
6	12.17	16.02	8.13	0.11	85.14	5.36	68.42	11.27
7	11.95	16.24	6.96	0.0	86.22	5.20	69.05	11.39
8	11.69	16.52	7.00	0.0	86.14	5.17	69.20	12.21

Table III

COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN COORDINATE
STATE VECTOR CONSTANT-VELOCITY MODELS

STATE VECTOR	CONSTANT VELOCITY TRAJECTORIES			MANEUVERING TRAJECTORIES		
	SHORT-RANGE	LONG-RANGE		SHORT-RANGE	LONG-RANGE	
	%hit %scare	%hit %scare		%hit %scare	%hit %scare	
SPHERICAL	86.89	5.00	40.81 21.14	25.34 24.05	0.71 0.95	
COORDINATE						
CARTESIAN	88.4	3.00	34.7 24.7	15.20 13.30	0.33 1.10	
COORDINATE						

trajectory. The results tabulated for the Cartesian coordinate state vector are the best of the results reported by Parr [Ref. 4].

Tables IV and V show the tracking performance of each optimal filter against its specific trajectory. For example, the tracking performance of filter 1 against trajectory 1 is given, and so on. The tabulated data represent the means of estimation error, defined by Equation (2.7), time averaged over the entire trajectory, and the RMS value of the deviation of the means of estimation error about their time average.

C. SIMULATION RESULTS FOR THE CORRELATED RANDOM ACCELERATION MODEL

The results presented in this section are for the optimized filter configurations. Since the optimization search for the ninth-order filter required a great deal more computer time, this model was only optimized against trajectories 1-3. As in the preceding section, the term "Filter 1A" denotes the filter with the specific $\underline{Q}(K)$ matrix that gave the best position prediction performance against trajectory 1.

The tabulated data of Table VI gives the elements of the $\underline{Q}(K)$ matrices that provided the optimum performance against each test trajectory. The data appear to be consistent within the tracking environments presented to the filter. Notice that in each configuration the elements of the optimum $\underline{Q}(K)$ matrices for the correlated random

Table IV
POSITION ESTIMATION PERFORMANCE OF THE
OPTIMIZED CONSTANT-VELOCITY MODEL FILTERS

FILTER	\bar{e}_r (yds)	$\sigma_{e_r}^-$ (yds)	\bar{e}_ϕ (deg)	$\sigma_{e_\phi}^-$ (deg)	\bar{e}_θ (deg)	$\sigma_{e_\theta}^-$ (deg)
1	5.33×10^{-2}	0.126	1.52×10^{-4}	6.18×10^{-4}	-2.67×10^{-4}	8.11×10^{-4}
2	-8.70×10^{-2}	0.282	2.32×10^{-3}	1.06×10^{-2}	2.90×10^{-6}	1.50×10^{-3}
3	4.95×10^{-2}	0.126	1.52×10^{-4}	6.18×10^{-4}	-2.68×10^{-4}	8.11×10^{-4}
4	-7.00×10^{-2}	0.219	1.69×10^{-3}	1.32×10^{-2}	5.22×10^{-6}	1.15×10^{-3}
5	9.04×10^{-2}	0.985	2.35×10^{-4}	1.95×10^{-2}	1.47×10^{-5}	1.56×10^{-3}
6	2.93×10^{-2}	0.894	-2.85×10^{-2}	2.93×10^{-1}	-3.77×10^{-4}	9.46×10^{-4}
7	6.00×10^{-2}	0.127	1.40×10^{-4}	6.21×10^{-4}	-2.50×10^{-4}	8.10×10^{-4}
8	5.25×10^{-2}	0.125	1.39×10^{-4}	6.19×10^{-4}	-2.60×10^{-4}	8.10×10^{-4}

Table V

RATE ESTIMATION PERFORMANCE OF THE
OPTIMIZED CONSTANT-VELOCITY MODEL FILTERS

FILTER	$\bar{e}_i^{\cdot} (y/s)$	$\sigma_{e_i^{\cdot}}^- (y/s)$	$\bar{e}_\phi^{\cdot} (d/s)$	$\sigma_{e_\phi^{\cdot}}^- (d/s)$	$\bar{e}_\theta^{\cdot} (d/s)$	$\sigma_{e_\theta^{\cdot}}^- (d/s)$
1	3.79	29.70	1.00×10^{-5}	5.26×10^{-4}	-2.45×10^{-4}	1.57×10^{-3}
2	3.69	29.72	3.36×10^{-2}	1.90×10^{-1}	-5.45×10^{-4}	5.71×10^{-3}
3	3.80	29.70	9.97×10^{-6}	5.26×10^{-4}	-2.45×10^{-4}	1.57×10^{-3}
4	3.70	29.71	3.75×10^{-3}	6.67×10^{-2}	-1.48×10^{-4}	2.18×10^{-3}
5	4.99	38.75	-2.51×10^{-2}	2.36×10^{-1}	-1.62×10^{-4}	5.77×10^{-3}
6	4.98	38.75	-1.15×10^{-2}	2.86×10^{-1}	-2.97×10^{-4}	1.68×10^{-3}
7	4.96	38.74	6.76×10^{-5}	5.27×10^{-4}	-2.38×10^{-4}	1.57×10^{-3}
8	4.97	38.73	6.67×10^{-5}	5.27×10^{-4}	-2.42×10^{-4}	1.57×10^{-3}

Table VI

DIAGONAL ELEMENTS OF THE OPTIMIZED
Q(K) MATRICES FOR THE CORRELATED
RANDOM ACCELERATION MODEL FILTERS

FILTER	q_{11}	q_{22}	q_{33}
1A	1.00×10^{-7}	5.00×10^{-9}	5.00×10^{-9}
2A	0.100	14.599	14.599
3A	1.00×10^{-3}	1.46×10^{-7}	1.46×10^{-7}

acceleration model are larger in magnitude than those for the corresponding entries of the constant-velocity model. This seems reasonable because this filter is required to respond to a transient (due to the non-zero acceleration rate) that is expected to be smaller in magnitude than the transient encountered by the constant-velocity model filter. For this reason the bandwidth of this filter must be increased to ensure sufficient response to the non-zero acceleration rates. During the optimization search, it was observed that, with one exception, this filter model is extremely sensitive to variations of elements q_{22} and q_{33} corresponding to acceleration rates normal to the Line-of-Sight. This is to be expected, since such accelerations cause the greatest displacement from the nominal trajectory adding a source for potentially large errors in the predicted position. The single exception noted was for trajectory 1. In this specific case the filter was surprisingly insensitive to variations in the elements of the $Q(K)$ matrix, and its predicted position response was fairly flat over a four-decade range (10^{-11} - 10^{-7}). The elements reported for Filter 1A are therefore somewhat arbitrary, lying in the center of the flat region. Values for the pair $q_{22} = q_{33}$ above and below this region produced inferior results. A refined search grid in the flat region failed to produce any significant change in the results reported for Filter 1A.

Table VII gives the simulation results for each optimal filter configuration and for each optimal filter configuration against the two remaining trajectories. Table VIII compares the hit/scare percentages from the simulations of the models using the two different state vectors. The data presented for the spherical coordinate state vector represent the optimum performance for the trajectories considered. The results reported for the Cartesian Coordinate state vector model represent the best of the results reported by Parr [Ref. 4]. Tables IX through XI show the tracking performance for each optimal filter configuration against its specific trajectory. The results shown are the time-averaged means of estimation error and the RMS value of the deviation of the means of estimation error from their time average.

Table VII

SIMULATION RESULTS FOR THE OPTIMIZED
CORRELATED RANDOM ACCELERATION MODELS

FILTER	TRAJECTORY 1		TRAJECTORY 2		TRAJECTORY 3	
	%hit	%score	%hit	%score	%hit	%score
1A	74.41	7.27	7.20	8.27	8.95	19.48
2A	32.89	15.57	26.43	14.89	0.0	0.05
3A	74.09	7.55	7.16	8.27	12.93	18.52

Table VIII

COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN

COORDINATE STATE VECTOR CORRELATED RANDOM ACCELERATION MODELS

STATE VECTOR	CONSTANT-VELOCITY TRAJECTORIES				MANEUVERING TRAJECTORIES			
	SHORT-RANGE		LONG-RANGE		SHORT-RANGE		LONG-RANGE	
	%hit	%scare	%hit	%scare	%hit	%scare	%hit	%scare
SPHERICAL	74.41	7.27	12.93	18.52	26.43	14.89	NOT TESTED	
COORDINATE								
CARTESIAN	73.40	7.70	6.60	17.40	21.20	9.00	NOT TESTED	
COORDINATE								

Table IX
POSITION ESTIMATION PERFORMANCE OF THE OPTIMIZED
CORRELATED RANDOM ACCELERATION MODEL FILTERS

PARAMETER	FILTER		
	1A	2A	3A
\bar{e}_r (yds)	5.87×10^{-2}	0.111	1.06×10^{-2}
$\sigma_{\bar{e}_r}$ (yds)	0.125	0.289	0.135
\bar{e}_ϕ (deg)	-5.45×10^{-5}	6.59×10^{-5}	2.55×10^{-4}
$\sigma_{\bar{e}_\phi}$ (deg)	7.55×10^{-4}	2.37×10^{-3}	8.23×10^{-4}
\bar{e}_θ (deg)	1.48×10^{-4}	2.41×10^{-4}	1.94×10^{-4}
$\sigma_{\bar{e}_\theta}$ (deg)	7.43×10^{-4}	1.40×10^{-3}	7.34×10^{-4}

Table X
RATE ESTIMATION PERFORMANCE OF THE OPTIMIZED
CORRELATED RANDOM ACCELERATION MODEL FILTERS

PARAMETER	FILTER		
	1A	2A	3A
$\bar{e}_{\dot{r}}$ (y/s)	3.77	3.83	3.78
$\sigma_{\bar{e}_{\dot{r}}}$ (y/s)	29.71	29.70	29.70
$\bar{e}_{\dot{\phi}}$ (d/s)	2.20×10^{-4}	1.64×10^{-2}	5.06×10^{-3}
$\sigma_{\bar{e}_{\dot{\phi}}}$ (d/s)	1.76×10^{-3}	0.109	1.39×10^{-2}
$\bar{e}_{\dot{\theta}}$ (d/s)	-1.58×10^{-4}	-1.50×10^{-4}	1.63×10^{-3}
$\sigma_{\bar{e}_{\dot{\theta}}}$ (d/s)	3.31×10^{-3}	8.14×10^{-3}	2.67×10^{-3}

Table XI
ACCELERATION ESTIMATION PERFORMANCE OF THE OPTIMIZED
CORRELATED RANDOM ACCELERATION MODEL FILTERS

PARAMETER	FILTER		
	1A	2A	3A
$\bar{e}_{\ddot{r}} (y/s^2)$	-0.546	-0.549	-0.526
$\sigma_{\bar{e}_{\ddot{r}}} (y/s^2)$	3.32	3.39	3.17
$\bar{e}_{\ddot{\phi}} (d/s^2)$	-4.35×10^{-4}	1.82×10^{-2}	-1.12×10^{-3}
$\sigma_{\bar{e}_{\ddot{\phi}}} (d/s^2)$	2.68×10^{-3}	0.31	1.39×10^{-2}
$\bar{e}_{\ddot{\theta}} (d/s^2)$	-9.70×10^{-4}	0.49	5.39×10^{-4}
$\sigma_{\bar{e}_{\ddot{\theta}}} (d/s^2)$	1.52×10^{-3}	0.10	2.67×10^{-3}

IV. ADAPTIVE FILTERING

A. THE ADAPTIVE BANDWIDTH FILTER

A single fixed-parameter filter configuration cannot realistically be expected to perform well against all trajectory profiles likely to be encountered. One way of approaching this problem is to use a filter that adapts its characteristics to a changing tracking environment, e.g., a target maneuver is detected. The design problem now becomes two-fold; under what circumstances, or when, should the filter's characteristics be changed and by how much should the filter be changed? Price and Brown [Ref. 11] discuss some interesting approaches to the design of adaptive filters; unfortunately their complexity is beyond the defined scope of this thesis.

The single method to be considered here will be a simple switch-on-range scheme. The motivation for such a scheme lies in the simplicity of its implementation. Additionally, Parr reported [Ref. 4] that of the methods he investigated, the switch-on-range method provided the best overall performance in his simulations. When filter tracking is initiated, a low $\tilde{Q}(K)$ gain schedule (corresponding to a narrow bandwidth filter) is used to provide the optimum performance against long-range targets. When the target's measured range reaches a predetermined switching point, the filter switches to a high $\tilde{Q}(K)$ gain

schedule (corresponding to a wideband filter) for the short-range tracking situation.

Both filter models developed in Chapter II are simulated in a switch-on-range adaptive mode of operation. Again, to duplicate Parr's testing environment, a switching range of 3333.33 yards is used, and the filters are tested against trajectories 9 - 14. The profiles for trajectories 9 - 14 are identical to Parr's trajectories 1 - 6. The results reported for the adaptive simulations are the products of 80-member ensemble Monte Carlo simulations. For the simulations of the constant-velocity model, the low $Q(K)$ gain schedule used was that from Filter 3. The data in Table II show that Filter 3 gave the best overall performance against the long-range trajectories treated in Chapter III. Similar considerations yield Filter 4 as the source for the high $Q(K)$ gain schedule for the short-range tracking environment. When filter tracking is initiated, the Kalman gains used are those for Filter 3. When the target's measured range becomes less than or equal to the switching range of 3333.33 yards, the Kalman gain schedule is reset to the steady-state gain values obtained from the gain schedule for Filter 4. Similar consideration of the data presented in Table VII gives Filter 2A as the source of the high $Q(K)$ gain schedule while Filter 3A provides the low $Q(K)$ gain schedule. As in the above simulation, the low $Q(K)$ gain schedule is used until the

switching range (3333.33 yards) is reached. At that point, the Kalman gain schedule is reset to the steady-state gain values from gain schedule for Filter 2A. Initialization of the filters was the same for all simulations conducted and is described by Equations (2.5), (2.8), and (2.9). The statistics assumed in all simulations for the measurement noise process $\tilde{v}(K)$ are given by Equations (1.6) and (3.1).

B. SIMULATION RESULTS FOR THE CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

The constant-velocity model filter was tested in a switch-on-range adaptive mode of operation against trajectories 9 - 14. Table XII shows the filter's position prediction accuracy results. The data tabulated in Table XIII compare the results for the adaptive simulations of the models using the two different state vectors. Tables XIV and XV show the tracking performance of the filter in the adaptive mode. As in Chapter III the tabulated data represent the time-averaged means of estimation error and the RMS value of the deviation of the means of estimation error from their time average.

C. SIMULATION RESULTS FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

The correlated random acceleration model was also tested in an adaptive mode of operation against trajectories 9 - 14. Table XVI shows this filter's position prediction performance results. Tables XVII - XIX show the filter's

Table XII
 SWITCH-ON-RANGE ADAPTIVE FILTER SIMULATION
 FOR THE CONSTANT-VELOCITY MODEL

TRAJECTORY	%HIT	%SCARE
9	90.58	5.55
10	60.57	12.98
11	71.58	11.39
12	63.95	15.45
13	67.37	9.98
14	74.67	9.65

Table XIII

COMPARISON OF RESULTS FOR THE SPHERICAL AND
CARTESIAN COORDINATE STATE VECTOR CONSTANT-
VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE
FILTERS

STATE VECTOR

TRAJECTORY	SPHERICAL		CARTESIAN	
	%HIT	%SCARE	%HIT	%SCARE
9	90.58	5.55	81.80	11.60
10	60.57	12.98	47.20	10.60
11	71.58	11.39	60.10	14.20
12	63.95	15.45	52.30	13.40
13	67.37	9.98	60.20	15.00
14	74.67	9.67	54.70	13.10

Table XIV

POSITION ESTIMATION PERFORMANCE FOR THE CONSTANT-VELOCITY MODEL
SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	\bar{e}_r (yds)	$\sigma_{\bar{e}_r}$ (yds)	\bar{e}_ϕ (deg)	$\sigma_{\bar{e}_\phi}$ (deg)	\bar{e}_θ (deg)	$\sigma_{\bar{e}_\theta}$ (deg)
9	6.34×10^{-2}	0.105	1.31×10^{-4}	8.59×10^{-4}	-1.18×10^{-5}	9.01×10^{-4}
10	-2.82×10^{-3}	0.261	1.52×10^{-2}	6.52×10^{-2}	3.59×10^{-6}	9.16×10^{-4}
11	-3.30×10^{-2}	0.228	4.60×10^{-3}	3.46×10^{-2}	-4.79×10^{-5}	1.03×10^{-3}
12	-1.30×10^{-2}	0.220	1.12×10^{-4}	2.30×10^{-2}	-1.18×10^{-5}	9.30×10^{-4}
13	-0.131	0.478	1.14×10^{-4}	4.33×10^{-2}	-1.18×10^{-5}	9.75×10^{-4}
14	-0.167	0.509	6.14×10^{-3}	7.85×10^{-2}	-1.2×10^{-4}	1.75×10^{-3}

Table XV

RATE ESTIMATION PERFORMANCE FOR THE CONSTANT-VELOCITY
MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	\bar{e}_r (y/s)	$\sigma_{e_r}^-$ (y/s)	\bar{e}_ϕ (d/s)	$\sigma_{e_\phi}^-$ (d/s)	\bar{e}_θ (d/s)	$\sigma_{e_\theta}^-$ (d/s)
9	1.27	15.81	6.94×10^{-5}	1.16×10^{-3}	-1.14×10^{-4}	1.50×10^{-3}
10	1.26	15.81	1.75×10^{-3}	9.92×10^{-2}	-1.46×10^{-4}	1.93×10^{-3}
11	1.24	15.81	1.96×10^{-2}	0.165	-1.24×10^{-3}	6.84×10^{-3}
12	1.25	15.81	-1.97×10^{-3}	0.109	-2.28×10^{-4}	2.40×10^{-3}
13	1.22	15.82	-6.48×10^{-3}	0.216	-1.87×10^{-4}	3.71×10^{-3}
14	1.18	15.82	1.47×10^{-2}	0.453	-1.92×10^{-3}	1.56×10^{-2}

tracking performance in the adaptive mode. As in the preceding section, the tabulated data represent the time-averaged means of estimation error and the RMS value of the deviation of the means of estimation error from their time average.

Table XVI
 SWITCH-ON-RANGE ADAPTIVE FILTER SIMULATION
 FOR THE CORRELATED RANDOM ACCELERATION MODEL

TRAJECTORY	%hit	%score
9	62.27	10.85
10	46.72	15.20
11	60.46	10.76
12	53.34	13.06
13	56.67	9.36
14	56.08	11.23

Table XVII

POSITION ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM
ACCELERATION SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	\bar{e}_r (yds)	$\sigma_{e_r}^-$ (yds)	\bar{e}_ϕ (deg)	$\sigma_{e_\phi}^-$ (deg)	\bar{e}_θ (deg)	$\sigma_{e_\theta}^-$ (deg)
9	3.64×10^{-2}	0.12	1.57×10^{-4}	1.13×10^{-3}	1.81×10^{-4}	1.11×10^{-3}
10	1.01×10^{-2}	0.20	-1.42×10^{-2}	5.69×10^{-2}	1.96×10^{-4}	1.12×10^{-3}
11	-1.24×10^{-3}	0.16	1.66×10^{-4}	1.49×10^{-3}	1.82×10^{-4}	1.11×10^{-3}
12	1.57×10^{-2}	0.17	1.60×10^{-4}	1.84×10^{-3}	1.81×10^{-4}	1.11×10^{-3}
13	-2.10×10^{-2}	0.31	1.57×10^{-4}	3.98×10^{-3}	1.81×10^{-4}	1.11×10^{-3}
14	-7.32×10^{-2}	0.32	-1.58×10^{-5}	5.01×10^{-3}	1.85×10^{-4}	1.13×10^{-3}

Table XVIII

RATE ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM
ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	$\bar{e}_r(y/s)$	$\sigma_{e_r}^-(y/s)$	$\bar{e}_\phi(d/s)$	$\sigma_{e_\phi}^-(d/s)$	$\bar{e}_\theta(d/s)$	$\sigma_{e_\theta}^-(d/s)$
9	1.26	15.81	2.16×10^{-4}	4.95×10^{-3}	2.87×10^{-4}	5.16×10^{-3}
10	1.26	15.82	2.70×10^{-3}	0.11	2.56×10^{-4}	5.32×10^{-3}
11	1.24	15.82	2.76×10^{-3}	4.44×10^{-2}	-7.30×10^{-4}	7.87×10^{-3}
12	1.25	15.82	-1.82×10^{-3}	2.03×10^{-2}	1.73×10^{-4}	5.36×10^{-3}
13	1.24	15.82	-6.35×10^{-3}	5.70×10^{-2}	2.14×10^{-4}	5.79×10^{-3}
14	1.20	15.84	-9.02×10^{-3}	0.20	-1.15×10^{-3}	1.15×10^{-2}

Table XIX

ACCELERATION ESTIMATION PERFORMANCE FOR THE CORRELATED
RANDOM ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	$\bar{e}_{\ddot{r}}(y/s^2)$	$\sigma_{\ddot{r}}(y/s^2)$	$\bar{e}_{\ddot{\phi}}(d/s^2)$	$\sigma_{\ddot{\phi}}(d/s^2)$	$\bar{e}_{\ddot{\theta}}(d/s^2)$	$\sigma_{\ddot{\theta}}(d/s^2)$
9	-1.09	1.72	3.70×10^{-4}	1.23×10^{-2}	-4.94×10^{-2}	3.99×10^{-2}
10	-0.112	1.73	-3.50×10^{-3}	0.236	-4.84×10^{-3}	2.76
11	0.108	1.73	-3.07×10^{-2}	0.566	-0.266	3.37
12	-0.112	1.73	-1.01×10^{-2}	0.111	-1.44×10^{-2}	2.99
13	-0.119	1.81	-1.98×10^{-2}	0.263	7.91×10^{-2}	6.34

V. SUMMARY AND CONCLUSIONS

A. SUMMARY OF TEST RESULTS

1. Summary of Simulation Results for the Two Models

The results presented in Chapter III show that against non-maneuvering target profiles the optimized constant-velocity models provided superior prediction performance. When the bandwidth of the filter is increased, adequate position prediction performance against maneuvering targets is obtained but at the expense of performance degradation against the non-maneuvering profiles. The optimized correlated random acceleration model showed only a slight improvement against the maneuvering profiles considered, but produced results inferior to those obtained from the constant-velocity model against the non-maneuvering target profiles. The performance of the constant-velocity model against long-range maneuvering profiles was significantly lower than for the short-range maneuvering profiles. This is to be expected since the long-range profiles require that the predictions be made over a longer period of time during which the target may maneuver unexpectedly and significantly. The tabulated data presenting the tracking performance of the filters developed in Chapter II, show that the position estimation performance for both filters is approximately the same. The correlated random acceleration model gave only

slightly better rate estimation performance than did the constant-velocity model. Against the target profiles considered the filter models derived from a spherical coordinate state vector provided significantly better overall predicted position results than the filter models derived from a Cartesian Coordinate state vector.

2. Summary of Simulation Results for the Adaptive Filters

The results reported in Chapter IV for the simulation of the switch-on-range adaptive filters indicated that the constant-velocity model derived from a spherical coordinate state vector provided significantly better predicted position performance than the constant-velocity model filter derived from a Cartesian Coordinate state vector against all target profiles considered. Further, the results indicate that of the models derived from a spherical coordinate state vector, the constant-velocity model gave the best overall predicted position performance.

B. CONCLUSIONS

A filter derived from a spherical coordinate state vector with assumed linear system dynamics produces predicted position results potentially superior to those obtained from a filter derived from a Cartesian Coordinate state vector. The spherical coordinate state vector constant-velocity model provides better overall predicted

position results than the correlated random acceleration model derived from a spherical coordinate state vector. The reason for this is perhaps due to the manner in which the predicted position, $\hat{\tilde{x}}(K+N/K)$ is generated. For both models considered, the predicted position $\hat{\tilde{x}}(K+N/K)$ is formed by raising the state transition matrix $\phi(K, K-1)$, to be the N^{th} power as shown by Equation (C.2). In the case of the constant-velocity model such a method uses the current position and velocity estimates to generate the predicted position $\hat{\tilde{x}}(K+N/K)$. The correlated random acceleration model uses the current position, velocity and acceleration estimates to generate the predicted position. Thus, any error in the estimate of the target's acceleration state causes significant errors in the predicted position when propagated over the prediction interval. These acceleration induced prediction errors are not present when only the current estimates of position and velocity are used as in the case of the constant-velocity model.

A filter model derived from a spherical coordinate state vector can produce results potentially superior to those obtained from the corresponding Cartesian Coordinate state vector model in an adaptive filtering environment.

C. SUGGESTIONS FOR FURTHER INVESTIGATION

Since the constant-velocity model produced the best overall predicted position results, a complete analysis of the sensitivity of this model's predicted position

performance to the statistics assumed for the measurement noise process $\tilde{y}(K)$ and the sampling period T should be carried out. Additional sensitivity studies should also focus on allowing the diagonal elements of $Q(K)$ to vary independently, thus removing the constraint that $q_{22} = q_{33}$.

For the correlated random acceleration model, one area that warrants future investigation is the use of only the position and velocity estimates of the filter in generating the predicted position, and comparing the results with those obtained from the constant-velocity model filter. This would reduce the errors in the predicted position due to errors in the current estimate of the target's acceleration state.

Both models should be tested against a greater variety of target profiles, especially targets that present crossing profiles in which large bearing and elevation rates are generated, and targets that execute maneuvers while in level flight.

Finally, more sophisticated adaptive filtering schemes should be developed and investigated for the constant-velocity model. A prime candidate for such an investigation is the residual testing method reported by Price and Brown [Ref. 11].

APPENDIX A

MONTE CARLO SIMULATION PROGRAM

A.1 PROGRAM LISTING AND DESCRIPTION

The Monte Carlo Simulation Program used in this thesis was written by Dr. D. E. Kirk of the Naval Postgraduate School. It is extremely flexible and well documented to assist the potential user in data preparation and input, as well as the various options available. It should be pointed out that SUBROUTINE KILL and all statements in the main program that deal with miss distance data were written specifically for this thesis. They may be deleted with no loss of program flexibility or functional operation.

Computer system requirements vary with the simulation and the user's desires. All the computer work done for this thesis was performed on the IBM SYSTEM 360/67 at the Naval Postgraduate School. The program as currently dimensioned requires 520K bytes of storage. When redimensioned for the 60 data point trajectories, 320K bytes of storage are required. A 100-member ensemble simulation of a linear sixth-order filter with no on-line computation required, the trajectory read in, miss distance information desired, and all printing and plotting options invoked requires approximately 1.5 minutes of CPU time for execution. An additional 30 seconds (approximately) are required for the compile/link process. For the work done for this thesis

the program was pre-compiled (FORTRAN IV Level G) and stored and executed from a data cell. The link process in this case required approximately two seconds. If the same linear, sixth-order filter is again simulated but the printed and plotted output is suppressed (retaining only the miss distance data) 59 seconds of CPU time is required to execute the simulation. In the case of the linear ninth-order filters that were simulated, the storage requirements remain unchanged, as do the time requirements for the compile/link process. A 100-member ensemble simulation of the linear ninth-order filter with no on-line computation required, miss distance information desired, the trajectory read in and all printing and plotting options invoked requires approximately 3.0 minutes of CPU time for execution. If only the miss distance information is desired and all other output is suppressed, approximately 2.35 minutes of CPU time is required to execute the simulation.


```

1 THIS PROGRAM PERFORMS MONTE CARLO SIMULATION OF STATE ESTIMATORS
2 OF WHICH THE KALMAN FILTER IS ONE EXAMPLE. THERE ARE SEVERAL
3 CPTICNS AVAILABLE AS INDICATED IN THE DETAILLED COMMENTS BELOW.
4 IT SHOULD BE NOTED THAT ALL COMPUTATIONS OF GAINS USING THE
5 SUBROUTINE GAIN ARE PERFORMED IN DOUBLE PRECISION. THUS ALL
6 AFRAYS FOR USE IN "GAIN" MUST BE PREPARED ACCORDINGLY.
7
8 REAL *8 GAMMA, COVW, R, PHI, H, TEMP, TEMP1, TEMP2, FKKM1, G, PPK, Q, EI, PPKM10
9 1, MISSD
10 COMMON GAMMA(9,9), COVW(9,9), R(9,9), PHI(9,9), H(9,9), TEMP(9,9), TEMPI(9,9)
11 1(9,9), TEMP2(9,9), PPKM1(9,9), PPKM10(9,9), G(9,9), PPK(9,9), Q(9,9), EI(9,9)
12 29(9,9), MISSD(100,160), VAR(9,9,160), GKS(9,9,160), PPKS(9,9,160), XM(9,9,160)
13 360(9,9), ERR(9,9,160), GAMMAS(9,9), PHIS(9,9,160), XS(9,9,160), HS(9,9), SIG(9,9), SIGMCSP
14 4W(9,9), X(9,9), SIGXZ(9,9), XZMEAN(9,9), XHKK(9,9), XHKKM1(9,9), VIMP(9,9), Z(9,9), V(9,9), SMCSP
15 5IGV(9,9), XHATZ(9,9), XINT(160), N, NSAM, IQ, M, ITER, IN, ISTAT, K, ITRD, IXMCSP
16 6Z, IV, IW, IEST, ND, NHIT, NSCARE, NMIS, NFIRE
17 DIMENSION XP(160), YP(160), SMISS(100,160), TEMPA(160)
18
19 N=ORDER OF SYSTEM MODEL AND FILTER (DIMENSION OF X,XHAT)
20
21 M=NUMBER OF MEASUREMENTS (DIMENSION OF THE VECTOR Z)
22
23 IN=NUMBER OF INPUT RANDOM FORCING FCNS (=DIMENSION OF W)
24
25 NSAM=NUMBER OF TIME SAMPLES
26
27 NENS=NUMBER OF MEMBERS IN ENSEMBLE
28
29 READ (5,106) N,M,IN,NSAM,NENS
30
31 READ (5,107) ND
32 THE VALUE OF ND READ IN MUST EQUAL THE ROW (AND COLUMN) DIMENSION
33 SPECIFIED FOR THE SQUARE MATRIX "TEMPI", E.G. IF TEMPI(3,3) IS
34 SPECIFIED IN THE COMMON STATEMENT "ND" MUST BE EQUAL TO 3.
35
36 IG=-1 -- GAINS COMPUTED OFF-LINE AND READ IN
37 0 -- GAINS COMPUTED ONLY ONCE BEFORE STARTING MONTE CARLO
38 1 -- GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE
39
40 IFLR=0 -- R IS READ IN
41 IFLR.NE.0 -- R IS COMPUTED ON-LINE AT EACH TIME SAMPLE
42
43 IEST=0 -- STANDARD KALMAN FILTER EQUATION IS USED
44 IEST.NE.0 -- STD. KALMAN FILTER EQ. NOT USED
45
46
47
48

```

CCCCCCCC

CCCCCCCCCCCCCCCC C C CCCCCCCCCCCCCCCCCC


```

ISIGV=0 -- STD. DEVIATIONS OF MEASUREMENT NCISE TO BE READ IN (SP) MCSP 97
ISIGW=0 -- STD. DEVIATIONS OF RANCCM INPUTS TO BE READ IN (SP) MCSP 98
IXHZ=0 -- INITIAL VALUE OF THE PREDICTED VALUE XH(0/-1) TO MCSP 99
          BE READ IN (S.P.) MCSP 100
          MCSP 101
          MCSP 102
          MCSP 103
          MCSP 104
          MCSP 105
          MCSP 106
          MCSP 107
          MCSP 108
          MCSP 109
          MCSP 110
          MCSP 111
          MCSP 112
          MCSP 113
          MCSP 114
          MCSP 115
          MCSP 116
          MCSP 117
          MCSP 118
          MCSP 119
          MCSP 120
          MCSP 121
          MCSP 122
          MCSP 123
          MCSP 124
          MCSP 125
          MCSP 126
          MCSP 127
          MCSP 128
          MCSP 129
          MCSP 130
          MCSP 131
          MCSP 132
          MCSP 133
          MCSP 134
          MCSP 135
          MCSP 136
          MCSP 137
          MCSP 138
          MCSP 139
          MCSP 140
          MCSP 141
          MCSP 142
          MCSP 143
          MCSP 144

IC=1 -- ONE INITIAL CONDITION VALUE USED FOR THE STATE MCSP
        OTHERWISE THE VALUE OF X(0) IS GENERATED BY A RANDOM MCSP
        NUMBER GENERATOR CALLED BY "XZER0". MCSP

READ (5,1-11) IPHI,IH,IR,IPKKMO,IGAM,ISIGV,IXHZ,IC,IMISS,IFIR MCSP
IE,IMISSK,IMISSR,IMISSI,IMISSC MCSP

IMISS=0      FIRING AND MISS DISTANCE DATA ARE DESIRED.  OTHER- MCSP
              WISE NOT. MCSP

IFIRE=0      FIRING DATA AVERAGED OVER ENTIRE MONTE CARLO RUN IS MCSP
              PRINTED.  OTHERWISE NOT. MCSP

IMISSK=0     MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE ARE MCSP
              PLOTTED VS. K.  OTHERWISE NOT. MCSP

IMISSR=0     MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE ARE MCSP
              PLOTTED VS. THE TARGET SLANT RANGE.  OTHERWISE NOT. MCSP

IMISSI=0     MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE ARE MCSP
              PLOTTED VS. THE AVERAGE LENGTH OF THE PREDICTION MCSP
              INTERVAL.  OTHERWISE NOT. MCSP

IMISSC=0     MEANS AND STANDARD DEVIATIONS OF THE MISS DISTANCE MCSP
              ARE PRINTED OUT.  OTHERWISE NOT. MCSP

NHIT = 0
NMISS = 0
NSCARE = 0
IRCS = 0
NFIRE = 0
IMISSN = 0
ISCARE = 0
CALL CVFLOW
IW = 6195217
IV = 1936748
IX2 = 135769

THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE RUN AS
SPECIFIED BY THE USER'S FLAGS

```


69

[illegible]

C	34	DC 34 J=1, IN	289	MCSP
C	34	GAMMAS(I,J) = GAMMA(I,J)	290	MCSP
C			291	MCSP
C			292	MCSP
C			293	MCSP
C		WRITE (6,161)	294	MCSP
C		CALL MWRITE (GAMMA,N,IN)	295	MCSP
C			296	MCSP
C			297	MCSP
C			298	MCSP
C	35	IF (IPKKMO.NE.O) GO TO 36	299	MCSP
C		CALL MREAD (PKKM10,N,N)	300	MCSP
C		WRITE (6,162)	301	MCSP
C		CALL MWRITE (PKKM10,N,N)	302	MCSP
C			303	MCSP
C			304	MCSP
C			305	MCSP
C	36	IF (ISIGV.NE.O) GO TO 37	306	MCSP
C		CALL VREAD (SIGV,M)	307	MCSP
C		WRITE (6,163)	308	MCSP
C		CALL VWRITE (SIGV,M)	309	MCSP
C			310	MCSP
C	37	IF (ISIGW.NE.O) GO TO 38	311	MCSP
C		CALL VREAD (SIGW,IN)	312	MCSP
C		WRITE (6,164)	313	MCSP
C		CALL VWRITE (SIGW,IN)	314	MCSP
C			315	MCSP
C			316	MCSP
C	38	IF (IXHZ.NE.O) GO TO 39	317	MCSP
C		CALL VREAD (XHATZ,N)	318	MCSP
C		WRITE (6,165)	319	MCSP
C		CALL VWRITE (XHATZ,N)	320	MCSP
C			321	MCSP
C	39	IF (IC.EQ.1) GO TO 40	322	MCSP
C		IC.NE.1 MEANS THAT MEANS AND STD. DEVIATIONS OF THE INITIAL STATE	323	MCSP
C		VALUE MUST BE READ IN. OTHERWISE NOT READ IN.	324	MCSP
C		CALL MREAD (XZMEAN,N)	325	MCSP
C		WRITE (6,166)	326	MCSP
C		CALL VWRITE (XZMEAN,N)	327	MCSP
C			328	MCSP
C			329	MCSP
C			330	MCSP
C		CALL VREAD (SIGXZ,N)	331	MCSP
C		WRITE (6,167)	332	MCSP
C		CALL VWRITE (SIGXZ,N)	333	MCSP
C		GC TO 45	334	MCSP
C			335	MCSP
C			336	MCSP


```

C      WRITE (6,160)
C      CALL MWRITE (Q,N,N)
47  IF (IG.NE.-1) GO TO 49
C
C      DC 48 K=1,NSAM
C
C      DC 48 I=1,N
48  READ (5,169) (GKS(I,J,K),J=1,M)
C
C      GC TO 52
49  IF (IG.NE.0) GO TO 52
C
C      DO 51 K=1,NSAM
C      CALL GAIN
C
C      DC 51 I=1,N
C
C      DC 50 L=1,N
50  PKKS(I,L,K) = PKK(I,L)
C
C      DC 51 J=1,M
51  GKS(I,J,K) = G(I,J)
C
52  CCNTINUE
C      IF GAINS WERE TO BE READ IN (IG=-1) OR COMPUTED ONLY
C      ONCE(IG=0), THIS HAS NOW BEEN DONE
C
C      SET UP ARRAYS FOR COMPUTING STATISTICS
C
C      DC 54 I=1,NENS
C
C      DC 53 J=1,NSAM
53  MISSD(I,J) = 0.000
C
54  CCNTINUE
C
C      DC 55 K=1,NSAM
C      YP(K) = 0.0
C      XP(K) = 0.0
C      XINT(K) = 0.0
C      TEMP(K) = 0.0

```

MCSP 385
 MCSP 386
 MCSP 387
 MCSP 388
 MCSP 389
 MCSP 390
 MCSP 391
 MCSP 392
 MCSP 393
 MCSP 394
 MCSP 395
 MCSP 396
 MCSP 397
 MCSP 398
 MCSP 399
 MCSP 400
 MCSP 401
 MCSP 402
 MCSP 403
 MCSP 404
 MCSP 405
 MCSP 406
 MCSP 407
 MCSP 408
 MCSP 409
 MCSP 410
 MCSP 411
 MCSP 412
 MCSP 413
 MCSP 414
 MCSP 415
 MCSP 416
 MCSP 417
 MCSP 418
 MCSP 419
 MCSP 420
 MCSP 421
 MCSP 422
 MCSP 423
 MCSP 424
 MCSP 425
 MCSP 426
 MCSP 427
 MCSP 428
 MCSP 429
 MCSP 430
 MCSP 431
 MCSP 432


```

C      DC 55 J=1,N
C      XM(J,K) = 0.
C      ERR(J,K) = 0.
C
C      DC 55 L=1,N
C      VAR(J,L,K) = 0.
C
C      BEGIN MAIN ITERATION LOOP HERE
C
C      DC 62 ITER=1,NENS
C      IF (IC.EQ.1) GO TO 56
C      CALL XZERO
C
C      DC 57 I=1,N
C      56 57 XFKKM1(I) = XHATZ(I)
C
C      DC 61 K=1,NSAM
C      FCKM NOISY MEASUREMENT FROM TRUE STATE VALUE
C
C      DC 58 I=1,N
C      58 X(I) = XS(I,K)
C
C      CALL MEAS
C      GAIN IS NOT TO BE COMPUTED ON-LINE IF IG.NE.1
C      IF (IG.NE.1) GO TO 60
C
C      Q IS TO BE COMPUTED ON-LINE IF IFLQ.NE.0
C      IF (IFLQ.NE.0) CALL GON
C      R IS TO BE COMPUTED ON-LINE IF IFLR.NE.0
C      IF (IFLR.NE.0) CALL RON
C      CALL GAIN
C
C      DC 59 I=1,N
C      DC 59 J=1,M
C      59 GKSI(J,K) = G(I,J)
C

```

```

MCSP 433
MCSP 434
MCSP 435
MCSP 436
MCSP 437
MCSP 438
MCSP 439
MCSP 440
MCSP 441
MCSP 442
MCSP 443
MCSP 444
MCSP 445
MCSP 446
MCSP 447
MCSP 448
MCSP 449
MCSP 450
MCSP 451
MCSP 452
MCSP 453
MCSP 454
MCSP 455
MCSP 456
MCSP 457
MCSP 458
MCSP 459
MCSP 460
MCSP 461
MCSP 462
MCSP 463
MCSP 464
MCSP 465
MCSP 466
MCSP 467
MCSP 468
MCSP 469
MCSP 470
MCSP 471
MCSP 472
MCSP 473
MCSP 474
MCSP 475
MCSP 476
MCSP 477
MCSP 478
MCSP 479
MCSP 480

```



```

66 CCNTINUE
C
C      IF (IPRT.NE.0) GO TO 72
C      WRITE (6,172) THEORETICAL COVARIANCES OF ESTIMATION ERROR
C      IF ONE SET OF GAINS HAS BEEN USED.
C      IF (IG.NE.0) GO TO 69
C      WRITE (6,173)
C
C      DC 67 K=1,NSAM
C      WRITE (6,174) K
C
C      DC 67 I=1,N
C      67 WRITE (6,171) (GKS(I,J,K),J=1,M)
C
C      WRITE (6,175)
C
C      DC 68 K=1,NSAM
C      WRITE (6,176) K
C
C      DC 68 I=1,N
C      68 WRITE (6,171) (PKKS(I,J,K),J=1,N)
C
C      69 WRITE (6,181)
C      WRITE (6,177)
C      WRITE (6,178)
C
C      DC 70 K=1,NSAM
C      WRITE (6,180)
C
C      DC 70 I=1,N
C      70 WRITE (6,179) K,I,XM(I,K),ERR(I,K),VAR(I,I,K)
C
C      WRITE (6,181)
C      IF (ISTAT.EQ.0) GO TO 72
C      WRITE (6,182)
C
C      DC 71 K=1,NSAM
C      WRITE (6,183) K
C
C      DC 71 I=1,N
C      71 WRITE (6,171) (VAR(I,L,K),L=1,I)
C
C      WRITE (6,181)
C      72 IF (IPLT.NE.0) GO TO 88
C
C      DO 73 K=1,NSAM

```

MCSP 529
 MCSP 530
 MCSP 531
 MCSP 532
 MCSP 533
 MCSP 534
 MCSP 535
 MCSP 536
 MCSP 537
 MCSP 538
 MCSP 539
 MCSP 540
 MCSP 541
 MCSP 542
 MCSP 543
 MCSP 544
 MCSP 545
 MCSP 546
 MCSP 547
 MCSP 548
 MCSP 549
 MCSP 550
 MCSP 551
 MCSP 552
 MCSP 553
 MCSP 554
 MCSP 555
 MCSP 556
 MCSP 557
 MCSP 558
 MCSP 559
 MCSP 560
 MCSP 561
 MCSP 562
 MCSP 563
 MCSP 564
 MCSP 565
 MCSP 566
 MCSP 567
 MCSP 568
 MCSP 569
 MCSP 570
 MCSP 571
 MCSP 572
 MCSP 573
 MCSP 574
 MCSP 575
 MCSP 576


```

C      73 XP(K) = K
C      IF (IGPLT.NE.1) GO TO 76
C      DC 75 I=1,N
C      DC 75 J=1,M
C      DC 74 K=1,NSAM
C      74 YP(K) = GKS(I,J,K)
C      WRITE (6,181)
C      CALL FLCIF (XP,YP,NSAM,0)
C      75 WRITE (6,184) I,J
C      76 IF (ITHVPL.NE.1) GO TO 79
C      DC 78 I=1,N
C      DC 77 K=1,NSAM
C      77 YP(K) = PKKS(I,I,K)
C      WRITE (6,181)
C      CALL FLCIF (XP,YP,NSAM,0)
C      78 WRITE (6,185) I,I
C      79 IF (IMTFLT.NE.1) GO TO 82
C      DC 81 I=1,N
C      DC 80 K=1,NSAM
C      80 YP(K) = XM(I,K)
C      WRITE (6,181)
C      CALL FLCIF (XP,YP,NSAM,0)
C      81 WRITE (6,186) I
C      82 IF (ISMPLT.NE.1) GO TO 85
C      DC 84 I=1,N
C      DC 83 K=1,NSAM
C      83 YP(K) = ERP(I,K)
C      WRITE (6,181)
C      CALL FLCIF (XP,YP,NSAM,0)
C      84 WRITE (6,187) I,I

```

```

MCSP 577
MCSP 578
MCSP 579
MCSP 580
MCSP 581
MCSP 582
MCSP 583
MCSP 584
MCSP 585
MCSP 586
MCSP 587
MCSP 588
MCSP 589
MCSP 590
MCSP 591
MCSP 592
MCSP 593
MCSP 594
MCSP 595
MCSP 596
MCSP 597
MCSP 598
MCSP 599
MCSP 600
MCSP 601
MCSP 602
MCSP 603
MCSP 604
MCSP 605
MCSP 606
MCSP 607
MCSP 608
MCSP 609
MCSP 610
MCSP 611
MCSP 612
MCSP 613
MCSP 614
MCSP 615
MCSP 616
MCSP 617
MCSP 618
MCSP 619
MCSP 620
MCSP 621
MCSP 622
MCSP 623
MCSP 624

```



```

85 IF (ISVPLT.NE.1) GO TO 88
C
DC 87 I=1,N
C
DC 86 K=1,NSAM
86 YP(K) = VAR(I,I,K)
C
WRITE (6,181)
CALL FLCTP(XP,YP,NSAM,O)
87 WRITE (6,188) I
C
WRITE (6,181)
C
THE FOLLOWING SECTION PLOTS THE DESIRED MISS DISTANCE INFORMATION.
C
88 IF (IMISS.NE.O) GO TO 105
C
DC 90 K=1,NSAM
C
DC 89 J=1,NENS
89 SMISS(J,K) = MISSD(J,K)
C
90 CCNTINUE
C
DC 92 K=1,NSAM
C
DC 91 J=1,NENS
91 YP(K) = YP(K)+SMISS(J,K)
C
92 CCNTINUE
C
DC 93 K=1,NSAM
93 YP(K) = YP(K)/ENS
C
ENS1 = ENS-1.
C
DC 95 K=1,NSAM
C
DC 94 I=1,NENS
94 TEMPA(K) = TEMPA(K)+((SMISS(I,K)-YP(K))**2
C
95 TEMPA(K) = SQRT(TEMPA(K)/ENS1)
C
IF (IMISSK.NE.O) GO TO 97
C

```

MCSP 625
 MCSP 626
 MCSP 627
 MCSP 628
 MCSP 629
 MCSP 630
 MCSP 631
 MCSP 632
 MCSP 633
 MCSP 634
 MCSP 635
 MCSP 636
 MCSP 637
 MCSP 638
 MCSP 639
 MCSP 640
 MCSP 641
 MCSP 642
 MCSP 643
 MCSP 644
 MCSP 645
 MCSP 646
 MCSP 647
 MCSP 648
 MCSP 649
 MCSP 650
 MCSP 651
 MCSP 652
 MCSP 653
 MCSP 654
 MCSP 655
 MCSP 656
 MCSP 657
 MCSP 658
 MCSP 659
 MCSP 660
 MCSP 661
 MCSP 662
 MCSP 663
 MCSP 664
 MCSP 665
 MCSP 666
 MCSP 667
 MCSP 668
 MCSP 669
 MCSP 670
 MCSP 671
 MCSP 672


```

C
96 DC 96 K=1, NSAM
   XF(K) = K
   WRITE (6,181)
   CALL FLCTP(XP,YP,NSAM,0)
   WRITE (6,199)
   WRITE (6,181)
   CALL FLCTP(XP,TEMPA,NSAM,0)
   WRITE (6,200)
97 IF (IMISSR.NE.0) GO TO 99
C
98 DC 98 K=1, NSAM
   XP(K) = XS(1,K)
   WRITE (6,181)
   CALL FLCTP(XP,YP,NSAM,0)
   WRITE (6,201)
   WRITE (6,181)
   CALL FLCTP(XP,TEMPA,NSAM,0)
   WRITE (6,202)
99 IF (IMISSI.NE.0) GO TO 101
C
100 DC 100 K=1, NSAM
   XINT(K) = XINT(K)/ENS
C
   WRITE (6,181)
   CALL FLCTP(XINT,YP,NSAM,0)
   WRITE (6,203)
   WRITE (6,181)
   CALL FLCTP(XINT,TEMPA,NSAM,0)
   WRITE (6,204)
   WRITE (6,181)
C
C THE FOLLOWING SECTION PRINTS THE DESIRED FIRING AND MISS DISTANCE
C DATA.
101 IF (IFIRE.NE.0) GO TO 102
   IF (NFIRE.EQ.0) GO TO 104
   RDS = NFIRE/ENS
   IRDS = RDS*0.5
   HIT = NHIT/ENS
   IHIT = HIT*0.5
   SCARE = NSCARE/ENS
   ISCARE = SCARE*0.5
   AMISS = NMISS/ENS
   AMISSN = AMISS*0.5
   PERHIT = (HIT/RDS)*100.
   PERSCR = (SCARE/RDS)*100.
C
C
C
C

```



```

102 PERMIS = (AMISS/RDS)*100.
    WRITE (6,195) IRDS
    WRITE (6,196) IHIT,PERHIT
    WRITE (6,197) ISCAR,PERSCR
    WRITE (6,198) IMISSN,PERMIS
    IF (IMISSC.NE.O) GO TO 105
    WRITE (6,181)
    WRITE (6,205)

```

```

C      DC 103 K=1,NSAM
103   WRITE (6,206) K,YP(K),TEMPA(K)

```

```

C      GC TO 105
104   WRITE (6,207)
105   CCNTINUE
    STCP

```

```

C      C
106   FFORMAT (5(110))
107   FFORMAT (12)
108   FFORMAT (7(110))
109   FFORMAT (2(15))
110   FFORMAT (5(110))
111   FFORMAT (1515)
112   FFORMAT (20X,'DESCRIPTION OF RUN',//)
113   FFORMAT (10X,'GAINS COMPUTED OFF-LINE AND READ IN',//)
114   FFORMAT (10X,'GAINS COMPUTED ONCE IN "GAIN" BEFORE STARTING MONTE CARLO',//)
115   FFORMAT (10X,'GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE',//)
116   FFORMAT (10X,'THE STANDARD LINEAR EQS. DO NOT CHARACTERIZE THE FILTER',//)
117   FFORMAT (10X,'THE STD. KALMAN EQS. CHARACTERIZE THE LINEAR FILTER',//)
118   FFORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS GENERATED BY SUBROUTINE TRACK',//)
119   FFORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS READ IN',//)
120   FFORMAT (10X,'SEVERAL TRACKS USED BUT NOT GENERATED FROM STD. LINEAR',//)
121   FFORMAT (10X,'SEVERAL TRACKS GENERATED BY USING THE STD. LINEAR DIFF',//)
122   FFORMAT (10X,'MEAN OF TRACK, MEAN CF EST. ERROR AND COVARIANCE OF TRACK',//)
123   FFORMAT (10X,'MEAN OF TRACK, MEAN AND VARIANCES OF EST. ERROR ARE',//)
124   FFORMAT (10X,'THE G MATRIX IS COMPUTED ON-LINE AT EACH SAMPLE BY "Q',//)
125   FFORMAT (10X,'THE COVARIANCE OF W IS READ IN AND Q IS COMPUTED BY "MC

```



```

126 1CMAT" BEFORE STARTING MONTE CARLO",//)
127 1FCRMAI (10X,"THE Q MATRIX IS READ IN",//)
128 1FCRMAI (10X,"R IS COMPUTED ON-LINE AT EACH SAMPLE BY "RON",//)
129 1FCRMAI (10X,"R IS READ IN",//)
130 1FCRMAI (10X,"20X,"INPUT DATA CALLED FOR",//)
131 1FCRMAI (10X,"PHI MATRIX",//)
132 1FCRMAI (10X,"H MATRIX",//)
133 1FCRMAI (10X,"R MATRIX",//)
134 1FCRMAI (10X,"C VARIANCE OF W",//)
135 1FCRMAI (10X,"Q MATRIX",//)
136 1FCRMAI (10X,"GAMMA MATRIX",//)
137 1FCRMAI (10X,"STANDARD DEVIATIONS OF MEASUREMENT NOISE",//)
138 1FCRMAI (10X,"STANDARD DEVIATIONS OF INPUT FORCING W",//)
139 1FCRMAI (10X,"XHAT(0/-1)",//)
140 1FCRMAI (10X,"MEANS AND VARIANCES OF X(0)",//)
141 1FCRMAI (10X,"P(0/-1)",//)
142 1FCRMAI (10X,"20X,"OUTPUT CALLED FOR",//)
143 1FCRMAI (10X,"PRINTED OUTPUT OF THE FOLLOWING DATA",//)
144 1FCRMAI (15X,"GAIN MATRICES AND THEORETICAL COVARIANCE OF EST. ERROR",//)
145 1FCRMAI (15X,"SAMPLE MEANS OF TRACK AND ESTIMATION ERROR, SAMPLE VARIANCES",//)
146 1FCRMAI (15X,"COVARIANCE OF TRACK VS. K",//)
147 1FCRMAI (10X,"NO PRINTED MONTE CARLO OUTPUT IS DESIRED",//)
148 1FCRMAI (15X,"G(K) VS. K",//)
149 1FCRMAI (15X,"P(K/K) THEORETICAL VS. K",//)
150 1FCRMAI (15X,"MEAN OF TRACK VS. K",//)
151 1FCRMAI (15X,"SAMPLE MEANS OF ESTIMATION ERROR VS. K",//)
152 1FCRMAI (15X,"SAMPLE VARIANCES OF ESTIMATION ERROR VS. K",//)
153 1FCRMAI (10X,"NO PLOTTED MONTE CARLO OUTPUT IS CALLED FOR",//)
154 1FCRMAI (20X,"INPUT DATA",//)
155 1FCRMAI (4X,"N=",12,4X,"M=",12,4X,"NSAM=",13,4X,"NENS=",1,4X,"IN=",12,4X,"ND=",12,4X,"//)
156 1FCRMAI (10X,"THE PHI MATRIX IS",//)
157 1FCRMAI (10X,"THE H MATRIX IS",//)
158 1FCRMAI (10X,"THE R MATRIX IS",//)
159 1FCRMAI (10X,"THE COVARIANCE OF W MATRIX IS",//)
160 1FCRMAI (10X,"THE Q MATRIX IS",//)
161 1FCRMAI (10X,"THE GAMMA MATRIX IS",//)
162 1FCRMAI (10X,"THE MATD. DEVIATIONS OF MEASUREMENT NOISE ARE",//)
163 1FCRMAI (10X,"THE STD. DEVIATIONS OF INPUT FORCING W ARE",//)
164 1FCRMAI (10X,"THE VECTOR OF THE VECTOR X(0) IS",//)
165 1FCRMAI (10X,"THE MEAN OF THE DEVIATIONS OF THE VECTOR X(0) ARE",//)
166 1FCRMAI (10X,"THE STANDARD STATE IS",//)
167 1FCRMAI (10X,"THE INITIAL STATE IS",//)
168 1FCRMAI (40IPE20.12))
169 1FCRMAI (40IPE20.12))

```



```

170 FCRMAT ((//,10X,'THE FIRST AND LAST POINTS ON THE SINGLE TRACK TO B',//) MCSP 817
171 E USED ARE,/) MCSP 818
172 FCRMAT ((9(2X,1PE12.5),//) DATA,///) MCSP 819
173 FCRMAT ((1,20X,'OUTPUT MATRICES ARE,/') MCSP 820
174 FCRMAT ((10X,'THE GAIN MATRICES ARE,/') MCSP 821
175 FCRMAT ((5X,'K=,13,10X,'G(K)=,/') MCSP 822
176 FCRMAT ((1X,10X,'THE THEORETICAL COVARIANCE MATRIX IS,/') MCSP 823
177 FCRMAT ((5X,'K=,13,10X,'P(K/K)=,/') MCSP 824
178 FCRMAT ((15X,'TIME,16,10X,'VECTOR CCM-,134,'SAMPLE MEAN', MCSP 825
179 FCRMAT ((15X,'SAMPLE MEAN OF,171,'SAMPLE VARIANCE CF,') MCSP 826
180 FCRMAT ((15X,'INDEX,16,10X,'PUNENT INDEX,134,'CF TRACK', MCSP 827
181 FCRMAT ((15X,'ESTIMATION ERROR,171,'ESTIMATION ERROR,') MCSP 828
182 FCRMAT ((6X,13,10X,1PE14.7,2(6X,1PE14.7)) MCSP 829
183 FCRMAT ((//,)) MCSP 830
184 FCRMAT ((10X,'THE SAMPLE COVARIANCE OF EST. ERROR MATRIX IS,/') MCSP 831
185 FCRMAT ((//,2X,'K=,13,') VS. K') MCSP 832
186 FCRMAT ((12X,'G(,11,') VS. K') MCSP 833
187 FCRMAT ((12X,'PKK(,11,') VS. K') MCSP 834
188 FCRMAT ((12X,'MEAN OF X(,11,') VS. K') MCSP 835
189 FCRMAT ((12X,'XHATKK(,11,') VS. K') MCSP 836
190 FCRMAT ((15X,'MEAN AND STANDARD DEVIATION OF MISS DISTANCE VS. K',//) MCSP 837
191 FCRMAT ((15X,'MEAN AND STANDARD DEVIATION OF MISS DISTANCE VS. SLAN',//) MCSP 838
192 FCRMAT ((15X,'MEAN AND STANDARD DEVIATION OF MISS DISTANCE VS. AVERAM',//) MCSP 839
193 FCRMAT ((15X,'MEAN AND STANDARD DEVIATION OF MISS DISTANCE VS. MISSEM',//) MCSP 840
194 FCRMAT ((15X,'MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE FOR MCSP 841
195 FCRMAT ((10X,'NO FIRING OR MISS DISTANCE DATA IS DESIRED,/') MCSP 842
196 FCRMAT ((15X,'AVERAGE NUMBER OF RCUNDS FIRED PER ENSEMBLE RUN =,13, MCSP 843
197 FCRMAT ((15X,'AVERAGE NUMBER OF HITS PER ENSEMBLE RUN =,13,2X,'OR', MCSP 844
198 FCRMAT ((15X,'AVERAGE NUMBER OF SCARES PER ENSEMBLE RUN =,13,2X,'CM', MCSP 845
199 FCRMAT ((15X,'AVERAGE NUMBER OF MAGGIES DRAWERS PER ENSEMBLE RUN =, MCSP 846
200 FCRMAT ((15X,'PERCENT,') VS. K',//) MCSP 847
201 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. K',//) MCSP 848
202 FCRMAT ((15X,'STANDARD DEVIATION OF MISS DISTANCE VS. K',//) MCSP 849
203 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. SLANT RANGE',//) MCSP 850
204 FCRMAT ((15X,'STANDARD DEVIATION OF MISS DISTANCE VS. SLANT RANGE',//) MCSP 851
205 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 852
206 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 853
207 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 854
208 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 855
209 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 856
210 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 857
211 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 858
212 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 859
213 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 860
214 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 861
215 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 862
216 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 863
217 FCRMAT ((15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTION',//) MCSP 864

```



```

204 FCRMAT (15X,'STANDARD DEVIATION CF MISS DISTANCE VS. AVERAGE LENGTH',MCSP 865
1H CF PREDICTION INTERVAL',//),MCSP 866
205 FCRMAT (14X,'K',5X,'MEAN OF MISS DIST.',2X,'STD. DEV. OF MISS DIST',MCSP 867
1',//),MCSP 868
206 FCRMAT (12X,'13,7X,1PE13.6,10X,1PE13.6,/')MCSP 869
207 FCRMAT (//,'15X,'NO ROUNDS WERE FIRED; GO DIRECTLY TO THE SHIPYARD',MCSP 870
1,DO NOT COLLECT KILL',//),MCSP 871
208 FCRMAT (10X,'THE FOLLOWING MISS DISTANCE DATA IS PLOTTED',//)MCSP 872
209 FCRMAT (10X,'NO PLOTTED MISS DISTANCE DATA IS CALLED FOR',//)MCSP 873
ENDMCSP 874

```



```

SUBROUTINE ESTIM
THIS SUBROUTINE UPDATES THE STATE ESTIMATE. IN THE DEFAULT
CCNDITION (IEST.EQ.0) THE STANDARD EQUATIONS
XHAT(K/K)=XHAT(K/K-1)+G(K)*(Z(K)-H(5)*XHAT(K/K-1))
XHAT(K+1/K)=PHI*XHAT(K/K)
ARE EVALUATED
REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10
1,MISSO
CCCOMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1(9,9),TEMP2(9,9),
1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(9,9),
29,9),MISSO(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,9,160),
360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIG(9,9),
4W(9,9),SIGXZ(9,9),XZMEAN(9,9),XHKK(9,9),XHKKM1(9,9),VTMP(9,9),V(9,9),SESTI
5IGV(9,9),XHATZ(9,9),XINT(160),N,NSAM,IG,M,ITER,IN,ISTAT,K,ITRO,IX
6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE
TAKE THE APPROPRIATE GAIN AND STORE IN THE ARRAY GK
DC 1 I=1,N
DC 1 J=1,M
1 GK(I,J) = GKS(I,J,K)
IF (IEST.NE.0) GO TO 2
CALL VPROD (HS,XHKKM1,M,N,VTMP)
CALL VSUB (Z,VTMP,M,VTMP)
CALL VPROD (GK,VTMP,N,M,VTMP)
CALL VADD (XHKKM1,VTMP,N,XHKK)
XHAT(K/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKK
CALL VPROD (PHIS,XHKK,N,N,XHKKM1)
XHAT(K+1/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKKM1
RETURN
2 CCATINUE
IF STANDARD EQUATIONS ARE NOT TO BE USED, THE APPROPRIATE
EQUATIONS MUST BE INSERTED HERE BY THE USER.
RETURN
END

```



```

SUBROUTINE GAIN
REAL *8 GAMMA, COVW, R, PHI, H, TEMP, TEMPI, TEMP2, PKKM1, G, PKK, Q, EI, PPKM10
1, MISSD
C COMMON GAMMA(9,9), COVW(9,9), R(9,9), PHI(9,9), H(9,9), TEMP(9,9), TEMPI(9,9),
1(9,9), TEMP2(9,9), PPKM1(9,9), PPKM10(9,9), G(9,9), PKK(9,9), Q(9,9), EI(9,9),
29,9), MISSD(100,160), VAR(9,9,160), GK(9,9,160), PKKS(9,9,160), XM(9,9,160),
360), ERR(9,160), GAMMAS(9,9), PHIS(9,9), XS(9,160), HS(9,9), GK(9,9), V(9,9), SIGGAIN
4W(9,9), X(9,9), SIGXZ(9,9), XZMEAN(9,9), XHKK(9,9), XHKKM1(9,9), VTMP(9,9), Z(9,9),
5IGV(9,9), XHATZ(9,9), XINT(160), N, NSAM, IQ, M, ITER, ITRK, IN, ISTAT, K, ITRO, IXGAIN
6Z, IV, IW, IEST, ND, NHIT, NSCARE, NMIS, NFIRE
C
      G(K) = P(K/K-1)*HT*(H*(P(K/K-1)*HT + R)
      IF (K.NE.1) GO TO 2
C
      DO 1 IL=1,N
C
        DO 1 JL=1,N
          1 PPKM1(IL,JL) = PPKM10(IL,JL)
C
        CC CONTINUE
          CALL TRANS (H,M,N,TEMP2)
          CALL PROD (PPKM1,TEMP2,N,M,TEMP)
          CALL PROD (H,TEMP,M,N,TEMP1)
          CALL ADD (TEMP1,R,M,M,TEMP1)
          IF (M.EQ.1) GO TO 4
          MD = ND
          CALL GAUSS3 (M,EPS,TEMP1,TEMP2,KER,MD)
          CALL PRCD (TEMP,TEMP2,N,M,M,G)
C
          NCTE HERE PPK(I,J) = P(K/K) WHERE
            P(K/K) = (I-G(K)*H)*P(K/K-1)
            CALL PROD (G,H,N,M,N,TEMP)
            CALL SUB (EI,TEMP,N,N,TEMP2)
            CALL PRCD (TEMP2,PPKM1,N,N,N,PKK)
C
          NCTE HERE PPKM1(I,J) = P(K/K-1) WHERE
            P(K/K-1) = PHI*(K-1/K-1)*PHIT + Q
            CALL TRANS (PHI,N,N,TEMP2)
            CALL PRCD (PPK,TEMP2,N,N,N,TEMP)
            CALL PRCD (PHI,TEMP,N,N,N,TEMP1)
            CALL ADD (TEMP1,Q,N,N,N,PPKM1)
          RETURN
C
          DC 5 I=1,N
          5 G(I,1) = TEMP(I,1)/TEMP1(I,1)

```


C

GO TO 3
END

GAIN 49
GAIN 50
GAIN 51
GAIN 52

[illegible]
$$Q = \text{GAMMA} * E(W * WT) * \text{GAMMAT}$$

```

REAL *8 GAMMA, COVW, R, PHI, H, TEMP, TEMP1, TEMP2, PKKM1, G, PKK, Q, EI, PKKM10
1, MISSD
1, COMMON GAMMA(9,9), COVW(9,9), R(9,9), PHI(9,9), H(9,9), TEMP(9,9), TEMP1(9,9), TEMP2(9,9), PKKM1(9,9), PKKM10(9,9), G(9,9), Q(9,9), EI(9,9), PKKM1(9,9), MISSD(100,160), VAR(9,9,160), GKS(9,9,160), PKKS(9,9,160), XM(9,9,160), ERR(9,160), GAMMAS(9,9), PHIS(9,9), XS(9,160), HS(9,9), GK(9,9), SIG(9,9), X(9), SIGXZ(9), XZMEAN(9), XHKK(9), XHKKM1(9), VTMP(9), Z(9), V(9), SIGV(9), XHAIZ(9), XINT(160), N, NSAM, IG, M, ITER, ITRK, IN, ISTAT, K, ITRO, IX, CMAT(9,9), IV, IW, IEST, ND, NHIT, NSCARE, NMIS$, NFIRE
6Z,
CALL PROD (GAMMA, COVW, N, IN, IN, TEMP)
CALL TRANS (GAMMA, N, IN, TEMP1)
CALL PRCD (TEMP, TEMP1, N, IN, N, G)
RETURN
END

```

```

CALL PROD (GAMMA,CCVW,N,IN,IN,TEMP)
CALL TRANS (GAMMA,N,IN,TEMP)
CALL PROD (TEMP,TEMP,N,IN,N,G)
RETURN
END

```



```

SUBROUTINE KILL
  REAL*8 GAMMA, COVW, R, PHI, H, TEMP, TEMPI, TEMP2, PKKM1, G, PKK, Q, EI, PKKM10
  1, MISSD, TEMPP1, PREPOS, VDTMP, ERI, DEGRAD, DELTAP, DELTR, D, CHORDI, AM2
  2, A, CHORD2, TEMPLEB, ELEG1, ERRNET, PI, DEGRAD, B, TEMPX, TEMPX1, SUM1, SUM2
  3, COMMON GAMMA(9,9), COVW(9,9), R(9,9), PHI(9,9), H(9,9), TEMP(9,9), TEMPI(9,9), EI(9,9), PKKM1(9,9), PKKM10(9,9), G(9,9), PKK(9,9), Q(9,9), XM(9,9), SIG(9,9), X(9,9), MISSD(100,160), GAMMA(9,9), PHIS(9,9), XKS(9,9,160), HS(9,9), GK(9,9), V(9,9), SKILL
  4, W(9,9), X(9,9), SIGXZ(9,9), XZMEAN(9,9), XHKK(9,9), XHKKM1(9,9), VIMP(9,9), Z(9,9), V(9,9), SKILL
  5, IGV(9,9), XHATZ(9,9), XINT(160), N, NSAM, IQ, M, ITRK, IN, ISTAT, K, ITR0, IX, KILL
  6, Z, IV, IW, IEST, ND, NHIT, NSCARE, NMIS, NFIRE
  7, DIMENSION TEMPP1(9,9), TEMPX(9,9), PREPOS(9,160), VDTMP(9), ERI(3), TEMP
  8, XI(9)
  9, DATA PI, DEGRAD, SAMPR, RING1, RING2/3.14159292635, 57.2957795131, 0.25,
  10, 18., 16./
  11, N1 = N/3
  12, N2 = 2*N1
  13, N3 = N1+1
  14, NN = N-1
  15,
  16,
  17,
  18,
  19,
  20,
  21,
  22,
  23,
  24,
  25,
  26,
  27,
  28,
  29,
  30,
  31,
  32,
  33,
  34,
  35,
  36,
  37,
  38,
  39,
  40,
  41,
  42,
  43,
  44,
  45,
  46,
  47,
  48,
  49,
  50,
  51,
  52,
  53,
  54,
  55,
  56,
  57,
  58,
  59,
  60,
  61,
  62,
  63,
  64,
  65,
  66,
  67,
  68,
  69,
  70,
  71,
  72,
  73,
  74,
  75,
  76,
  77,
  78,
  79,
  80,
  81,
  82,
  83,
  84,
  85,
  86,
  87,
  88,
  89,
  90,
  91,
  92,
  93,
  94,
  95,
  96,
  97,
  98,
  99,
  100,
  101,
  102,
  103,
  104,
  105,
  106,
  107,
  108,
  109,
  110,
  111,
  112,
  113,
  114,
  115,
  116,
  117,
  118,
  119,
  120,
  121,
  122,
  123,
  124,
  125,
  126,
  127,
  128,
  129,
  130,
  131,
  132,
  133,
  134,
  135,
  136,
  137,
  138,
  139,
  140,
  141,
  142,
  143,
  144,
  145,
  146,
  147,
  148,
  149,
  150,
  151,
  152,
  153,
  154,
  155,
  156,
  157,
  158,
  159,
  160,
  161,
  162,
  163,
  164,
  165,
  166,
  167,
  168,
  169,
  170,
  171,
  172,
  173,
  174,
  175,
  176,
  177,
  178,
  179,
  180,
  181,
  182,
  183,
  184,
  185,
  186,
  187,
  188,
  189,
  190,
  191,
  192,
  193,
  194,
  195,
  196,
  197,
  198,
  199,
  200,
  201,
  202,
  203,
  204,
  205,
  206,
  207,
  208,
  209,
  210,
  211,
  212,
  213,
  214,
  215,
  216,
  217,
  218,
  219,
  220,
  221,
  222,
  223,
  224,
  225,
  226,
  227,
  228,
  229,
  230,
  231,
  232,
  233,
  234,
  235,
  236,
  237,
  238,
  239,
  240,
  241,
  242,
  243,
  244,
  245,
  246,
  247,
  248,
  249,
  250,
  251,
  252,
  253,
  254,
  255,
  256,
  257,
  258,
  259,
  260,
  261,
  262,
  263,
  264,
  265,
  266,
  267,
  268,
  269,
  270,
  271,
  272,
  273,
  274,
  275,
  276,
  277,
  278,
  279,
  280,
  281,
  282,
  283,
  284,
  285,
  286,
  287,
  288,
  289,
  290,
  291,
  292,
  293,
  294,
  295,
  296,
  297,
  298,
  299,
  300,
  301,
  302,
  303,
  304,
  305,
  306,
  307,
  308,
  309,
  310,
  311,
  312,
  313,
  314,
  315,
  316,
  317,
  318,
  319,
  320,
  321,
  322,
  323,
  324,
  325,
  326,
  327,
  328,
  329,
  330,
  331,
  332,
  333,
  334,
  335,
  336,
  337,
  338,
  339,
  340,
  341,
  342,
  343,
  344,
  345,
  346,
  347,
  348,
  349,
  350,
  351,
  352,
  353,
  354,
  355,
  356,
  357,
  358,
  359,
  360,
  361,
  362,
  363,
  364,
  365,
  366,
  367,
  368,
  369,
  370,
  371,
  372,
  373,
  374,
  375,
  376,
  377,
  378,
  379,
  380,
  381,
  382,
  383,
  384,
  385,
  386,
  387,
  388,
  389,
  390,
  391,
  392,
  393,
  394,
  395,
  396,
  397,
  398,
  399,
  400,
  401,
  402,
  403,
  404,
  405,
  406,
  407,
  408,
  409,
  410,
  411,
  412,
  413,
  414,
  415,
  416,
  417,
  418,
  419,
  420,
  421,
  422,
  423,
  424,
  425,
  426,
  427,
  428,
  429,
  430,
  431,
  432,
  433,
  434,
  435,
  436,
  437,
  438,
  439,
  440,
  441,
  442,
  443,
  444,
  445,
  446,
  447,
  448,
  449,
  450,
  451,
  452,
  453,
  454,
  455,
  456,
  457,
  458,
  459,
  460,
  461,
  462,
  463,
  464,
  465,
  466,
  467,
  468,
  469,
  470,
  471,
  472,
  473,
  474,
  475,
  476,
  477,
  478,
  479,
  480,
  481,
  482,
  483,
  484,
  485,
  486,
  487,
  488,
  489,
  490,
  491,
  492,
  493,
  494,
  495,
  496,
  497,
  498,
  499,
  500,
  501,
  502,
  503,
  504,
  505,
  506,
  507,
  508,
  509,
  510,
  511,
  512,
  513,
  514,
  515,
  516,
  517,
  518,
  519,
  520,
  521,
```



```

C      IF(NSAMP-GE.K) GO TO 3
C      IF(EOF-LE.10.0) GO TO 31
C      ERRNET = 6.00 02
C      GO TO 17
C      31 CONTINUE
C      XINT(K) = XINT(K)+TNSAMP
C
C      DO 4 I=1,N
C      4 TEMPX(I) = PREPOS(I,K-NSAMP)
C
C      THIS SECTION RAISES PHI TO THE NSAMP POWER.
C
C      DO 5 I=1,N
C      5 DO 5 J=1,N
C      5 TEMPPI(I,J) = 0.000
C
C      TEMPPI(1,1) = 1.000
C      TEMPPI(1,2) = TNSAMP*SAMP
C      TEMPPI(2,2) = 1.000
C      IF(N-NE.9) GO TO 8
C      TEMPPI(3,3) = PHI(3,3)**NSAMP
C      SUM1 = 0.00
C      SUM2 = 0.00
C      NSAMP1 = NSAMP-1
C
C      DO 6 I=1,NSAMP
C      6 J=I-1
C      6 SUM1 = SUM1+(PHI(3,3)**J)
C
C      DO 7 I=1,NSAMP1
C      7 J=I-1
C      7 SUM2 = SUM2+(TNSAMP-1)*(PHI(3,3)**J)
C
C      TEMPPI(2,3) = PHI(2,3)*SUM1
C      TEMPPI(1,3) = PHI(1,3)*SUM1+PHI(1,2)*PHI(2,3)*SUM2
C      DO 10 NA=N1,N2,N1

```



```

C      DO 9 I=1,N1
C      INA = I+NA
C
C      DO 9 J=1,N1
C      JNA = J+NA
C      9  TEMPP1(INA,JNA) = TEMPP1(I,J)
C
C      1C CONTINUE
C
C      FORM THE PREDICTED POSITION, XHAT(K/K-NSAMP) = (PHI*NSAMP)*
C      XHAT(K-NSAMP/K-NSAMP)
C
C      DO 12 I=1,N
C
C      DO 11 J=1,N
C      11  VDTEMP(I) = VDTEMP(I)+TEMPP1(I,J)*TEMPX(J)
C
C      12 CCCONTINUE
C
C      COMPUTE THE PREDICTION ERRORS AND THE MAGNITUDE OF THE ERROR
C      VECTOR.
C
C      J = 0
C
C      DO 13 I=1,NN,N1
C      J = J+1
C      13  ERI(J) = TEMPX1(I)-VDTEMP(I)
C
C
C      DELTR = DABS(ERI(1))
C      DELTAP = DABS(ERI(2))
C      DELTAT = DABS(ERI(3))
C      C = DMAX1(TEMPX1(1),VDTEMP(1))*D
C      CHORD1 = 2.*DSIN(DELTAP/2.)
C      ANGLE1 = (PI-DELTAT)/2.
C      ELEGL = ((CHORD1**2)+(DELTR**2)-2.*DELTR*CHORD1*DCOS(ANGLEA))
C      ANGLEB = (PI-DELTAT)/2.
C      CHCRD2 = 2.*DSIN(DELTAT/2.)*D
C      B = DSQRT(ELEGL)
C      EFRNET = DSQRT(ELEGL+((CHORD2**2)-2.*B*(CHCRD2)*DCOS(ANGLEB))
C
C      COMPUTE ROUNDS FIRED, HITS, MISSES AND SCARES
C
C      IF (ERRNET.GT.RING2) GO TO 15

```



```

IF (ERRNET.LE.RING1) GO TO 14
NSCARE = NSCARE+1
GO TO 16
14 NHIT = NHIT+1
GO TO 16
15 NMISSE = NMISSE+1
16 NFIRE = NFIRE+1
17 MISSD(ITER,K) = ERRNET
RETURN
END

```

```

KILL 154
KILL 155
KILL 156
KILL 157
KILL 158
KILL 159
KILL 160
KILL 161
KILL 162
KILL 166

```



```

SUBROUTINE QONL
  IF Q IS TO BE COMPUTED ON-LINE (IFLQ.NE.0) IT IS DONE
  IN THIS SUBROUTINE

  REAL *8 GAMMA, COVW, R, PHI, H, TEMP, TEMP1, TEMP2, PKKM1, G, PKK, Q, EI, PKKM10
  1, MISSD
  COMMON GAMMA(9,9), COVW(9,9), R(9,9), PHI(9,9), H(9,9), TEMP(9,9), TEMP1(9,9), TEMP2(9,9), PKKM1(9,9), PKKM10(9,9), G(9,9), PKK(9,9), Q(9,9), EI(9,9), MISSD(100,160), VAR(9,9,160), GKS(9,9,160), PKKS(9,9,160), XM(9,9,160), ERR(9,160), GAMMAS(9,9), PHIS(9,9), XS(9,9,160), HS(9,9), GK(9,9), SIG(9,9), X(9,9), SIGXZ(9,9), XZMEAN(9,9), XHKK(9,9), XHKKM1(9,9), VTMP(9,9), Z(9,9), V(9,9), S(9,9), XHATZ(9,9), XINT(160), N, NSAM, IQ, M, ITER, ITRK, IN, ISTAT, K, ITRO, IX
  51, IGV(9,9), XHATZ(9,9), XINT(160), N, NSAM, IQ, M, ITER, ITRK, IN, ISTAT, K, ITRO, IX
  6Z, IV, IW, IEST, ND, NHIT, NSCARE, NMISS, NFIRE

  THE APPROPRIATE STATEMENTS FOR COMPUTING Q ON-LINE MUST
  BE INSERTED HERE BY THE USER

  RETURN
  END

```



```

SUBROUTINE RON
IF R IS TO BE COMPUTED ON-LINE (IFLR.NE.O) IT IS DONE
IN THIS SUBROUTINE

REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM1ORONL
1,MISSD
COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1RONL
1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(9,9)RONL
29(9,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XN(9,9),XRONL
360),EKR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGRONL
4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SRONL
5IGV(9),XHAIZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRC,IXRONL
6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE
THE APPROPRIATE STATEMENTS FOR COMPUTING R ON-LINE MUST
BE INSERTED HERE BY THE USER

RETURN
END

```



```

SUBROUTINE STAT
THIS SUBROUTINE COMPUTES RUNNING SUMS USED IN DETERMINING THE
SAMPLE STATISTICS OF TRACK AND ESTIMATION ERRORS. IN THE DEFAULT
OPTION (ISTAT.EQ.0) THE STATISTICS TO BE COMPUTED ARE MEAN OF
TRACK, MEAN OF ESTIMATION ERROR AND VARIANCE OF ESTIMATION
ERROR. IF (ISTAT.NE.0) THE OFF-DIAGONAL TERMS IN THE COVARIANCE OF
ESTIMATION ERROR MATRIX ARE ALSO COMPUTED.
REAL *8 GAMMA, COVW, R, PHI, H, TEMP, TEMP1, TEMP2, FKKM1, G, FKK, Q, EI, PKKM10
1, MISSD
1 CCMMON GAMMA(9,9), COVW(9,9), R(9,9), PHI(9,9), H(9,9), TEMP(9,9), TEMP1(9,9), TEMP2(9,9), FKKM10(9,9), PKK(9,9), Q(9,9), XM(9,9), EI(9,9), HS(9,9), GK(9,9), V(9,9), ITRK, IN, ISTAT, K, ITR0, IX
1(9,9), TEMP2(9,9), PKKM1(9,9), VAR(9,9,160), GKS(9,9,160), PKKS(9,9,160), XS(9,160), XHKKM1(9), XHKK(9), NSAM, IQ, M, ITR, ITRK, IN, ISTAT, K, ITR0, IX
29(9), MISSD(100,160), VAR(9,9,160), GKS(9,9,160), PKKS(9,9,160), XS(9,160), XHKKM1(9), XHKK(9), NSAM, IQ, M, ITR, ITRK, IN, ISTAT, K, ITR0, IX
360), ERR(9,160), GAMMA(9,9), PHIS(9,9), XS(9,160), HS(9,9), GK(9,9), V(9,9), ITRK, IN, ISTAT, K, ITR0, IX
4W(9), X(9), SIGX2(9), XZMEAN(9), XHKK(9), XHKKM1(9), VIMP(9), Z(9), V(9), ITRK, IN, ISTAT, K, ITR0, IX
51GV(9), XHAT2(9), XINT(160), N, NSAM, IQ, M, ITR, ITRK, IN, ISTAT, K, ITR0, IX
6Z, IV, IW, IEST, ND, NHIT, NSCARE, NMIS, NFIRE
DIMENSION EXH(9)
IF (ITRK.NE.1) GO TO 2
IF (ITER.NE.1) GO TO 4
DC 1 J=1,N
1 XM(J,K) = XS(J,K)
GO TO 4
2 CCNTINUE
DC 3 J=1,N
3 XM(J,K) = XM(J,K)+XS(J,K)
4 CCNTINUE
DC 5 J=1,N
EXH(J) = XHKK(J)-XS(J,K)
ERR(J,K) = ERR(J,K)+EXH(J)
5 VAR(J,J,K) = VAR(J,J,K)+EXH(J)**2
IF (ISTAT.EQ.0) RETURN
DO 6 L=2,N
LM1 = L-1

```



```

C
C
C      DC 6 J=1,LMI
6 VAR(L,J,K) = VAR(L,J,K)+EXH(L)*EXH(J)
      RETURN
      END

```

```

STAT 49
STAT 50
STAT 51
STAT 52
STAT 53
STAT 54
STAT 55

```


C	IF(ITRK.EC.1) THE USER MUST INSERT HERE THE STATEMENTS REQUIRED	TRCK	49
C	TO GENERATE A SINGLE TRAJECTORY AND STORE IT IN THE ARRAY IS TO BE	TRCK	50
C	XS(I,K), I=1,N, K=2, NSAM (NOTE THAT IF A SINGLE TRAJECTORY IS TO BE	TRCK	51
C	GENERATED, THE INITIAL CONDITION HAS BEEN READ IN AND STORED	TRCK	52
C	IN XS(I,1), I=1,N)	TRCK	53
C		TRCK	54
	DC 5 K=2, NSAM	TRCK	55
	EKM1 = K-1	TRCK	56
	T = 0.5*EK M1	TRCK	57
	XS(1,K) = 13.-0.0173205*T	TRCK	58
	XS(2,K) = -0.0173205	TRCK	59
	XS(3,K) = 7.7-0.01*T	TRCK	60
5	XS(4,K) = -0.01	TRCK	61
C		TRCK	62
	RETURN	TRCK	63
6	CONTINUE	TRCK	64
C	IF THIS POINT IS REACHED, ITRK NOT EQUAL 0 CR 1 INDICATING THAT	TRCK	65
C	SEVERAL TRACKS ARE TO BE GENERATED, BUT NOT BY USING THE STD.	TRCK	66
C	LINEAR DIFFERENCE EQS..THE USER MUST SUPPLY THE APPROPRIATE	TRCK	67
C	STATEMENTS HERE.	TRCK	68
	RETURN	TRCK	69
	END	TRCK	70
		TRCK	71
		TRCK	72

C	1	SUBROUTINE XZERO	XZER
C	2	THIS SUBROUTINE GENERATES THE INITIAL STATE VALUE FROM A NORMAL	XZER
C	3	RANDCM NUMBER GENERATOR. IT IS ASSUMED THAT THE INITIAL STATE	XZER
	4	HAS COMPONENTS THAT ARE INDEPENDENT	XZER
	5	REAL *8 GAMMA, COVW, R, PHI, H, TEMP, TEMPI, TEMP2, PKKM1, G, PKK, Q, EI, PKKM10	XZER
	6	1, MISSD	XZER
	7	COMMON GAMMA(9,9), COVW(9,9), R(9,9), PHI(9,9), H(9,9), TEMP(9,9), TEMP1	XZER
	8	1(9,9), TEMP2(9,9), PKKM1(9,9), PKKM10(9,9), G(9,9), PKK(9,9), Q(9,9), XM(9,9), EI	XZER
	9	29(9,9), MISSD(100,160), VAR(9,9,160), GKS(9,9,160), PKKS(9,9,160), HS(9,9), GK(9,9), SIG	XZER
	10	360), ERR(9,160), GAMMAS(9,9), PHIS(9,9), XS(9,160), XHKKM1(9), VITER, ITRK, IN, ISTAT, K, ITRO, I	XZER
	11	4W(9), X(9), SIGXZ(9), XZMEAN(9), XHKK(9), XHKKM1(9), VITER, ITRK, IN, ISTAT, K, ITRO, I	XZER
	12	5IGV(9), XHATZ(9), XINT(160), N, NSAM, IQ, M, NFIRE	XZER
	13	6Z, IV, IW, IEST, ND, NHIT, NSCARE, NMIS, NFIRE	XZER
	14	CALL SNCRM (IXZ, X, N)	XZER
	15		XZER
	16		XZER
	17		XZER
	18		XZER
	19		XZER
	20		XZER
	21		XZER
	22		XZER

C	1	DC 1 I=1,N
C	2	XS(I,I) = SIGXZ(I)*X(I)+XZMEAN(I)
C	3	RETURN
	4	END


```

SUBROUTINE MREAD (A,N,M)
THIS SUBROUTINE READS AN NX M MATRIX A ACCORCING TO THE FORMAT
8F10.0. THE ENTRIES IN THE FIRST ROW OF A ARE READ FIRST, THEN
THE ENTRIES IN THE SECOND ROW, AND SO ON.
REAL *8A
DIMENSION A(9,9)

```

```

      DO 1 I=1,N
      1 READ (5,2) (A(I,J),J=1,M)

```

```

      RETURN

```

```

      2 FORMAT (8F10.0)
      END

```

```

1 MRED
2 MRED
3 MRED
4 MRED
5 MRED
6 MRED
7 MRED
8 MRED
9 MRED
10 MRED
11 MRED
12 MRED
13 MRED
14 MRED
15 MRED
16 MRED
17 MRED

```



```

C      SUBROUTINE MWRITE (A,N,M)
C      THIS SUBROUTINE WRITES THE ENTRIES OF THE NXM MATRIX A
C      REAL *8A
C      DIMENSION A(9,9)
C
C      DO 1 I=1,N
C      1 WRITE (6,2) (A(I,J),J=1,M)
C
C      RETURN
C
C      2 FORMAT (9(2X,1PE12.5))
C      END

```

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE
MWRITE

```


C	C	1	MSUB
C	C	2	MSUB
C	C	3	MSUB
C	C	4	MSUB
C	C	5	MSUB
C	C	6	MSUB
C	C	7	MSUB
C	C	8	MSUB
C	C	9	MSUB
C	C	10	MSUB
C	C	11	MSUB
C	C	12	MSUB
C	C	13	MSUB
C	C	14	MSUB
C	C	15	MSUB
C	C	16	MSUB


```

SUBROUTINE SUB (A,B,N,M,C)
THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX
A AND STORES THE RESULT IN C
REAL *8 A,B,C
DIMENSION A(9,9), B(9,9), C(9,9)

DO 1 I=1,N
  DO 1 J=1,M
    1 C(I,J) = A(I,J)-B(I,J)

  RETURN
  END

```



```

C
C
C
C
C
SUBROUTINE VADD (X,Y,N,Z)
THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND
Y AND STORES THE RESULT IN THE N-VECTOR Z
REAL *4X(9),Y(9),Z(9)
      DC 1 I=1,N
      1 Z(I) = X(I)+Y(I)
      RETURN
      END
C
C

```

```

1 VADD
2 VADD
3 VADD
4 VADD
5 VADD
6 VADD
7 VADD
8 VADD
9 VADD
10 VADD
11 VADD
12 VADD
13 VADD

```



```

SUBROUTINE VPROD (A,X,M,N,Y)
THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX
A AND THE N-VECTOR X AND STORES THE RESULT IN THE
N-VECTOR Y

```

```

REAL *4A(9,9),X(9),Y(9),T(9)

```

```

DO 1 I=1,M
T(I) = 0.00

```

```

      DC 1 J=1,N
1 T(I) = T(I)+A(I,J)*X(J)

```

```

      DC 2 I=1,M
2 Y(I) = T(I)

```

```

RETURN
END

```

```

1 VPRD
2 VPRD
3 VPRD
4 VPRD
5 VPRD
6 VPRD
7 VPRD
8 VPRD
9 VPRD
10 VPRD
11 VPRD
12 VPRD
13 VPRD
14 VPRD
15 VPRD
16 VPRD
17 VPRD
18 VPRD
19 VPRD
20 VPRD
21 VPRD
22 VPRD
23 VPRD
24 VPRD
25 VPRD

```



```

C
C
C
C
SUBROUTINE VREAD (V,N)
THIS SUBROUTINE READS THE N-DIMENSIONAL S.P. VECTOR V
DIMENSION V(9)
READ (5,1) (V(I),I=1,N)
RETURN
C
C
1 FORMAT (8F10.0)
END

```

```

VRED
VRED
VRED
VRED
VRED
VRED
VRED
VRED
VRED
VRED
VRED
VRED

```



```

C C C
C C C
C C C
C C C
C C C
SUBROUTINE VSUB (X,Y,N,Z)
THIS SUBROUTINE COMPUTES THE DIFFERENCE X-Y OF THE TWO
N-VECTORS X & Y AND STORES THE RESULT IN THE N-VECTOR Z
REAL *4X(9),Y(9),Z(9)
      DC 1 I=1,N
      1 Z(I) = X(I)-Y(I)
      RETURN
      END
C C

```

```

VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB
VSUB

```


A.2 LISTING OF KEY VARIABLES USED IN THE MONTE CARLO PROGRAM

AMISS	average number of misses over the entire track (scalar).
COVW(m,m)	covariance of the system random forcing input matrix (double precision).
EI(n,n)	identity matrix (double precision).
ERR(N,nsam)	sample mean of estimation error vector (single precision).
G(n,m)	sample gain matrix (double precision).
GAMMA(n,m)	system random forcing input distribution matrix (double precision).
GAMMAS(n,m)	system random forcing input distribution matrix (single precision).
GK(n,m)	sample gain matrix (single precision).
GKS(n,m,nsam)	filter gain schedule (single precision).
H(m,n)	system observation matrix (double precision).
HIT	average number of hits over the entire track (scalar).
HS(m,n)	system observation matrix (single precision).
ITER	ensemble member index.
M	number of measurements.
MISSD(nens,nsam)	sample miss distance matrix (double precision).
N	system order.
NENS	number of members in the ensemble.
NSAM	number of time points to be processed.
PHI(n,n)	state transition matrix (double precision).

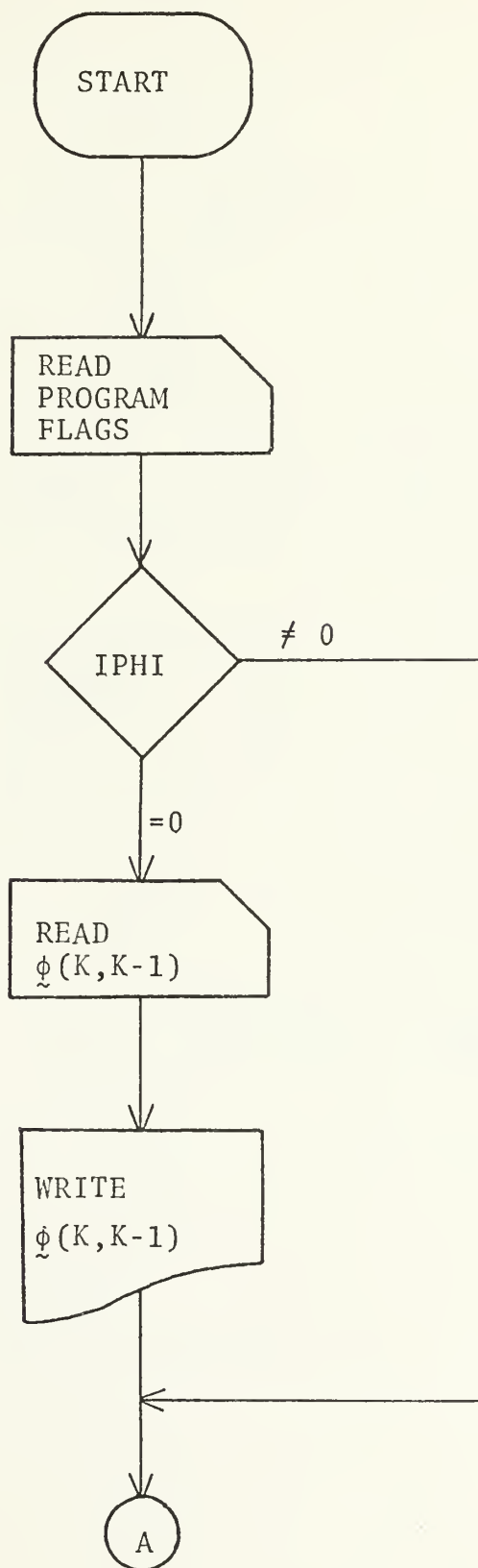
PHIS(n,n)	state transition matrix (single precision).
PKK(n,n)	theoretical covariance of estimation error matrix $\tilde{P}(K/K)$ (double precision).
PKKM1(n,n)	one-step prediction error covariance matrix $\tilde{P}(K+1/K)$ (double precision).
PKKM10(n,n)	initial value of the covariance of estimation error matrix $\tilde{P}(0/-1)$ (double precision).
PKKS(n,n,nsam)	sample covariance of estimation error matrix $\tilde{P}(K/K)$ (single precision).
Q(m,m)	system random forcing input covariance matrix (double precision).
R(m,m)	covariance of measurement noise matrix (double precision).
SCARE	average number of scares over the entire track.
SIGV(m)	standard deviation of measurement noise vector (single precision).
SIGW(m)	standard deviation of system random forcing input vector (single precision).
SIGXZ(n)	standard deviation of system initial condition vector (single precision).
V(m)	measurement noise vector (single precision).
VAR(n,n,nsam)	actual covariance of estimation error matrix (single precision).
W(m)	system random forcing input vector (single precision).
X(n)	state vector (single precision).
XHATZ(n)	initializing value of the state estimate vector $\hat{\tilde{x}}(0/-1)$ (single precision).
XHKK(n)	state estimate vector $\hat{\tilde{x}}(K/K)$ (single precision).
XHKKM1(n)	one-step state prediction vector $\hat{\tilde{x}}(K/K-1)$ (single precision).

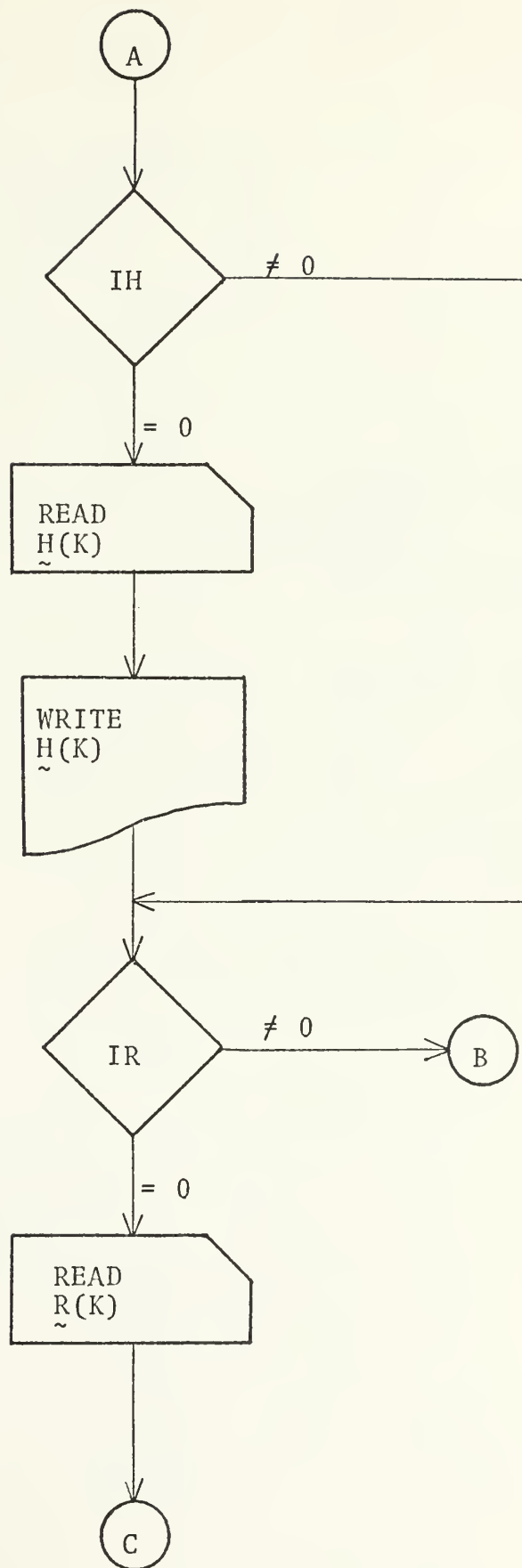
XM(n,nsam)	mean state trajectory over the ensemble (single precision).
XS(n,nsam)	reference trajectory (single precision).
XZMEAN(n)	mean of initial condition vector (single precision).

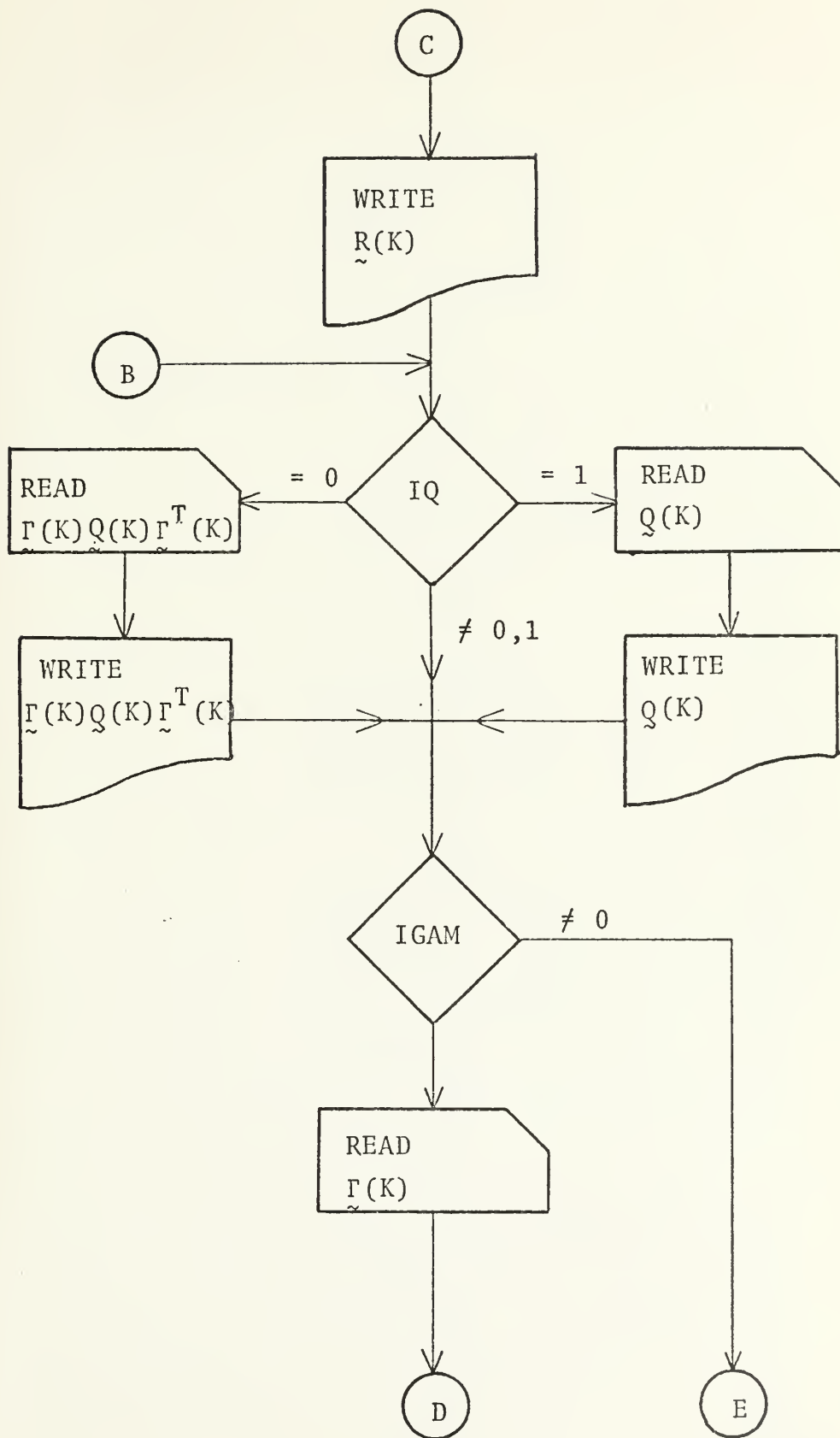
A.3 IBM SYSTEM 360/67 SOURCE LIBRARY SUBROUTINES CALLED
BY THE MONTE CARLO SIMULATION PROGRAM

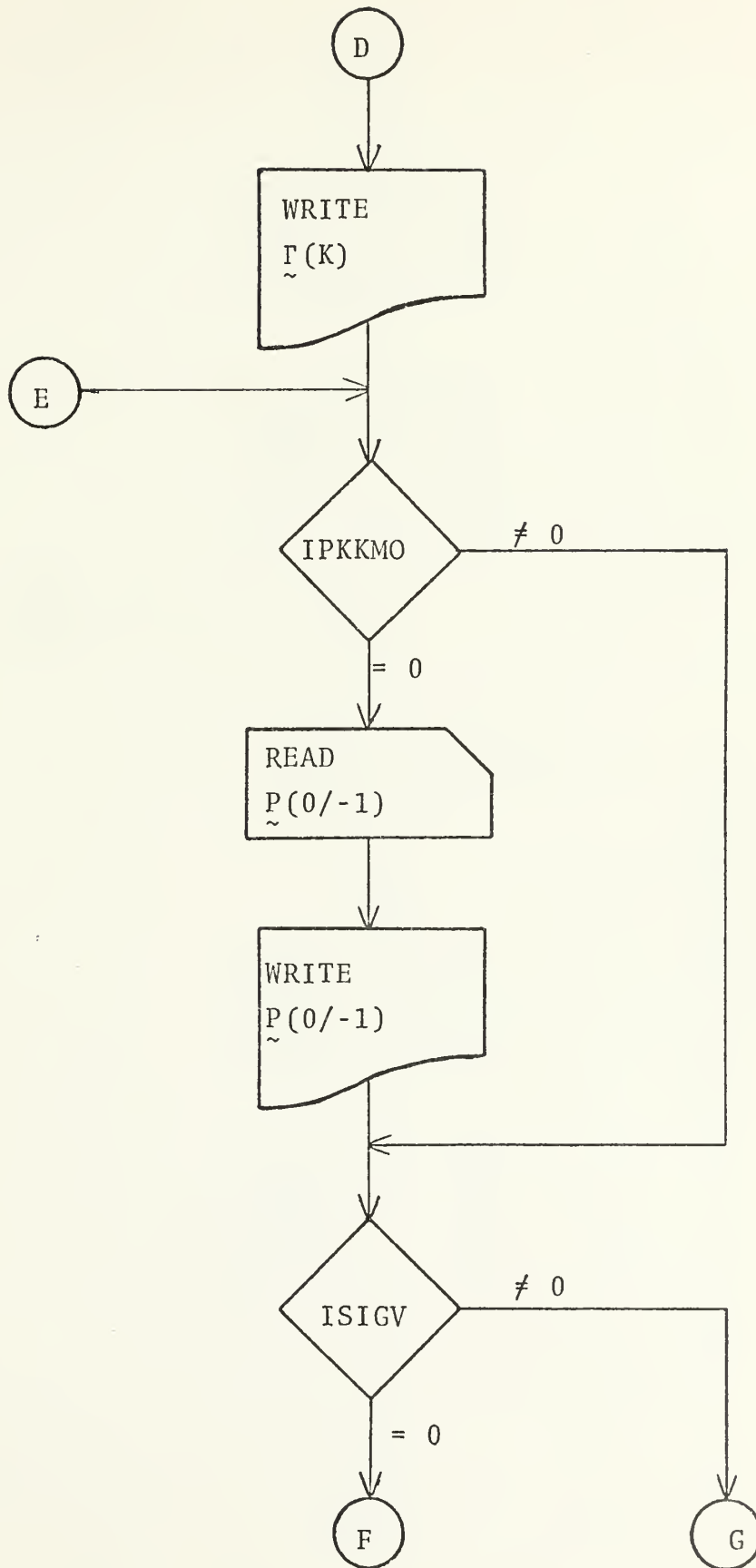
GAUSS3	matrix inversion by the method of GAUSS.
OVFLOW	utility program required by the random number generator.
PLOTP	generates two-dimensional line printer plots.
SNORM	random number generator that gives normally distributed, shuffled pseudo-random numbers.

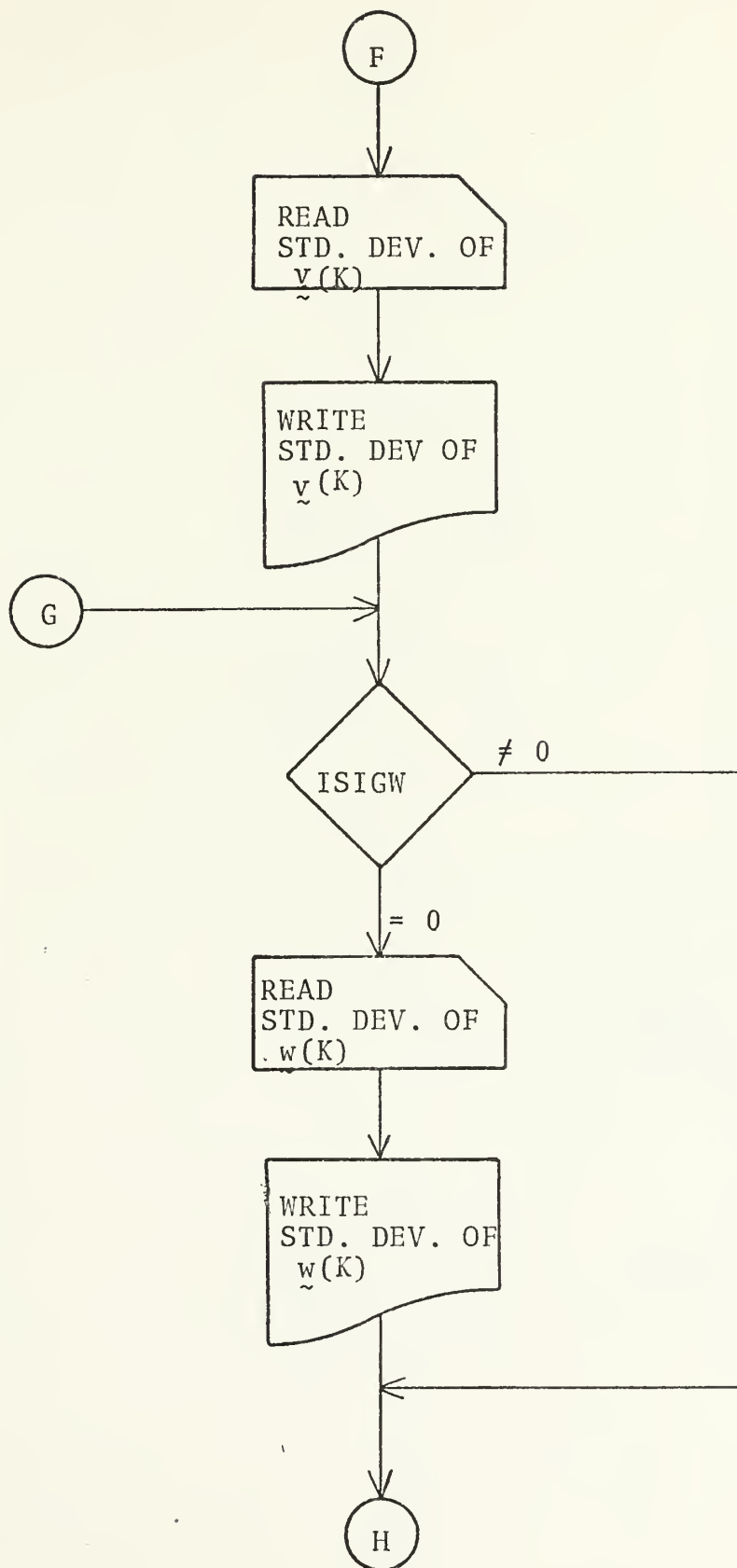
A.4 MONTE CARLO SIMULATION PROGRAM FLOWCHART

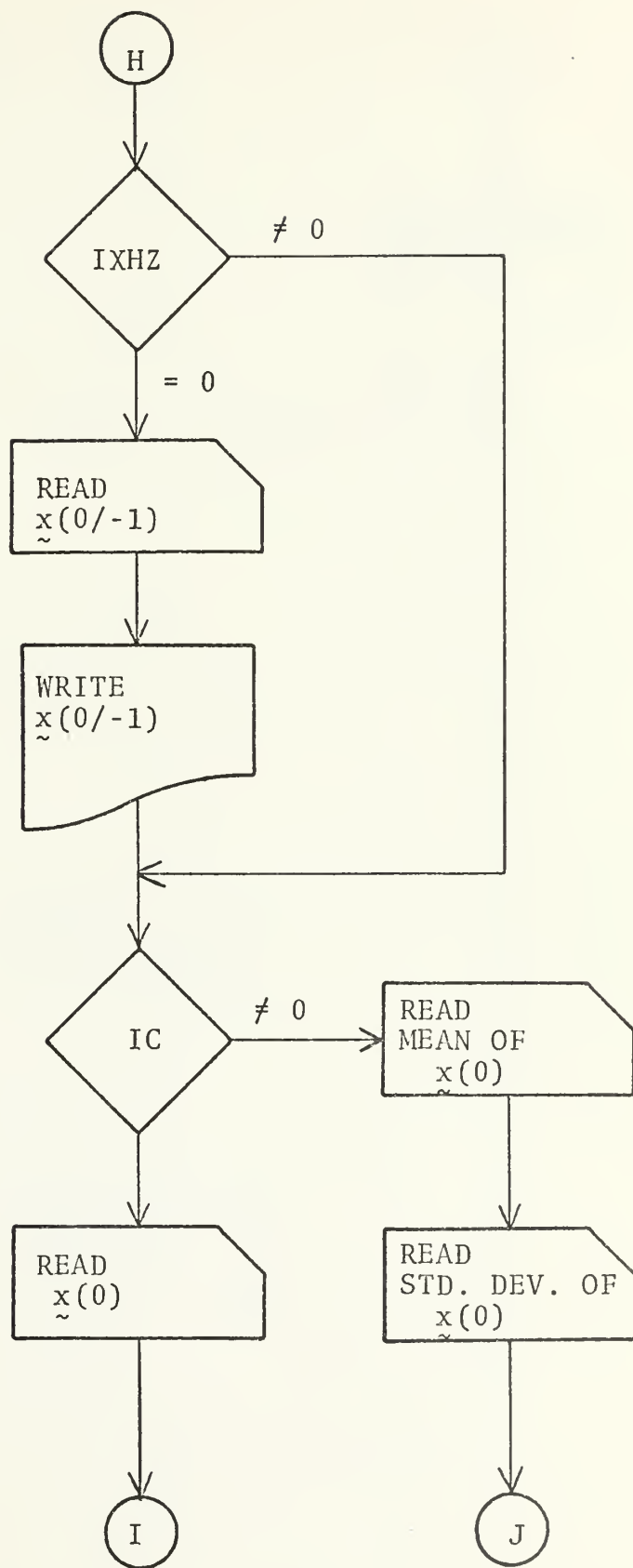


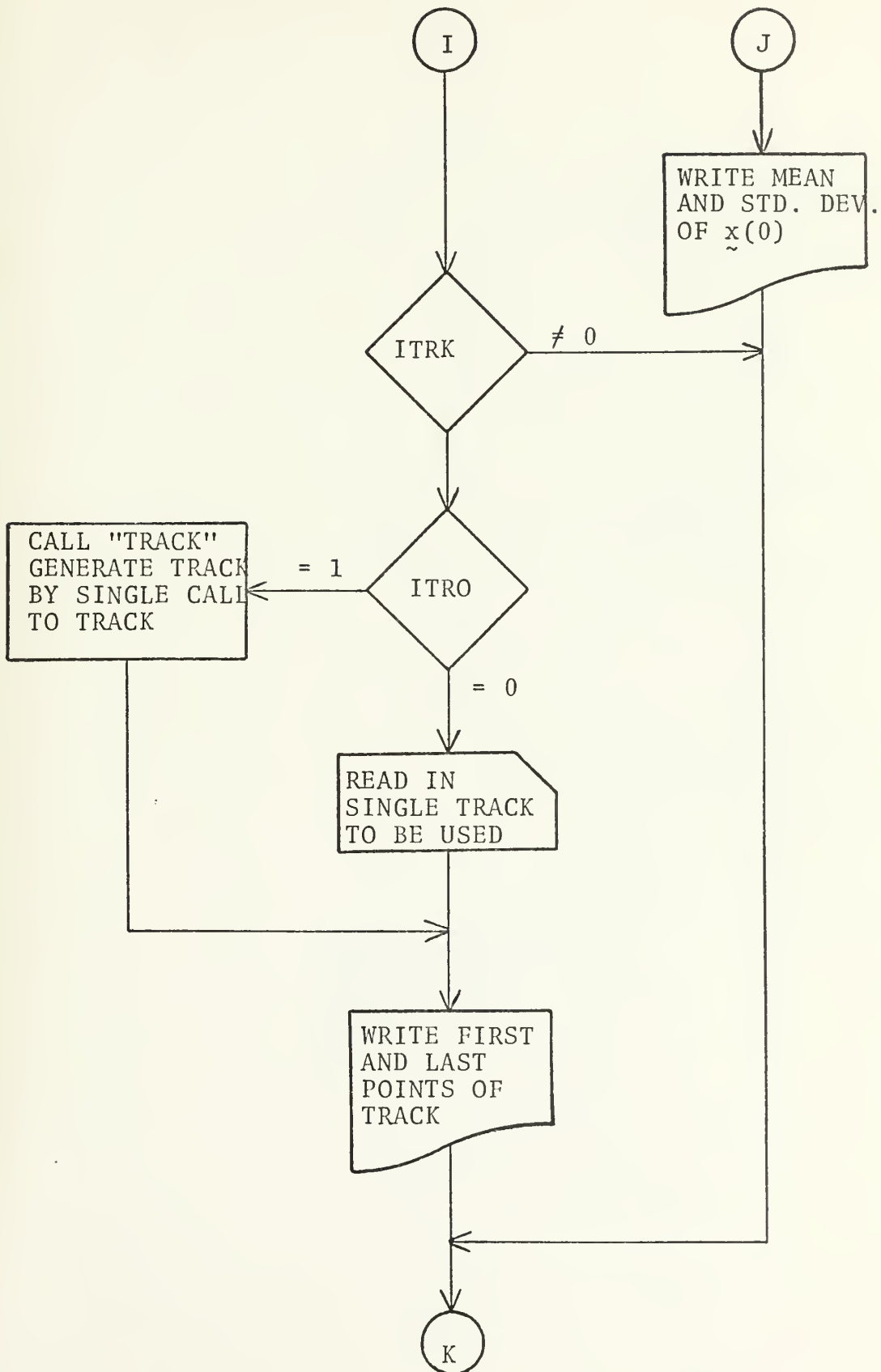


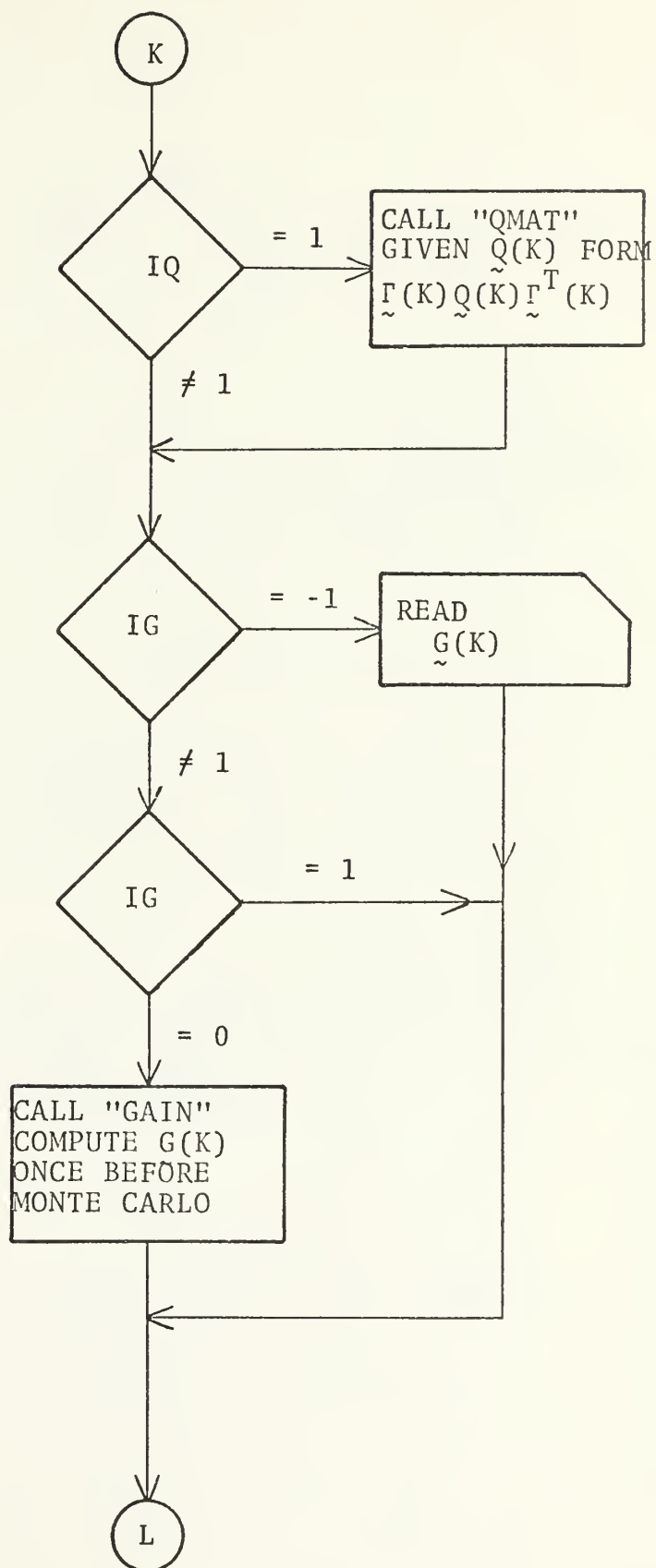


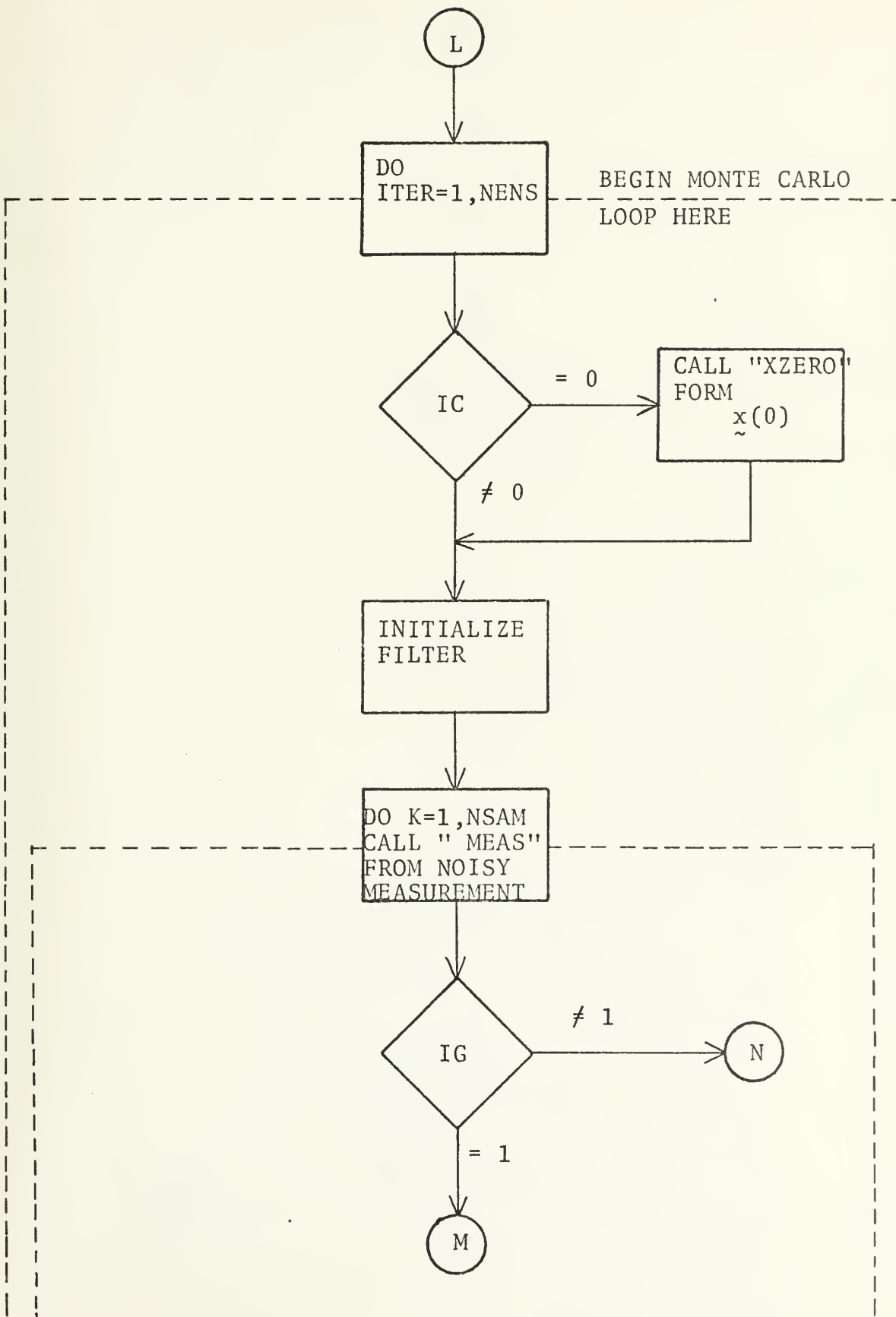


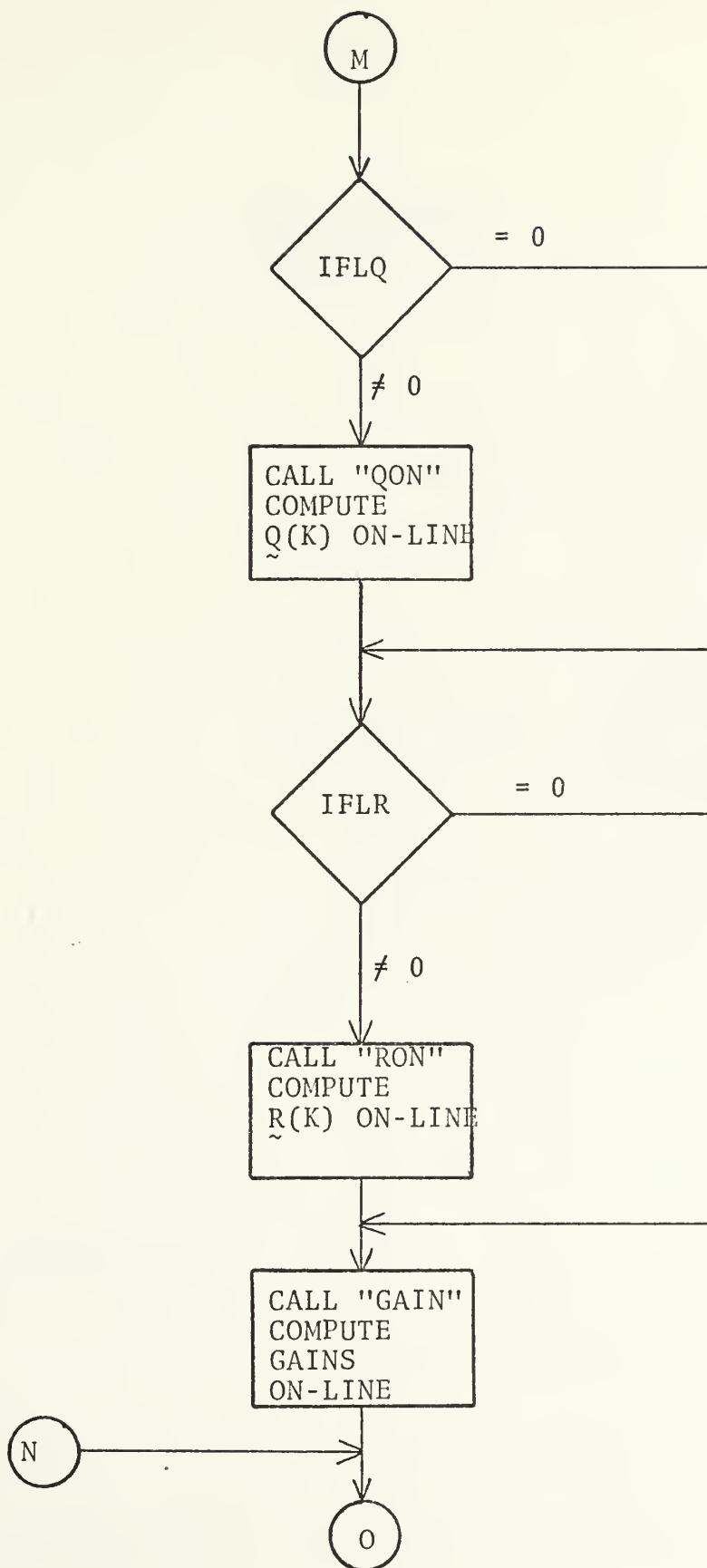


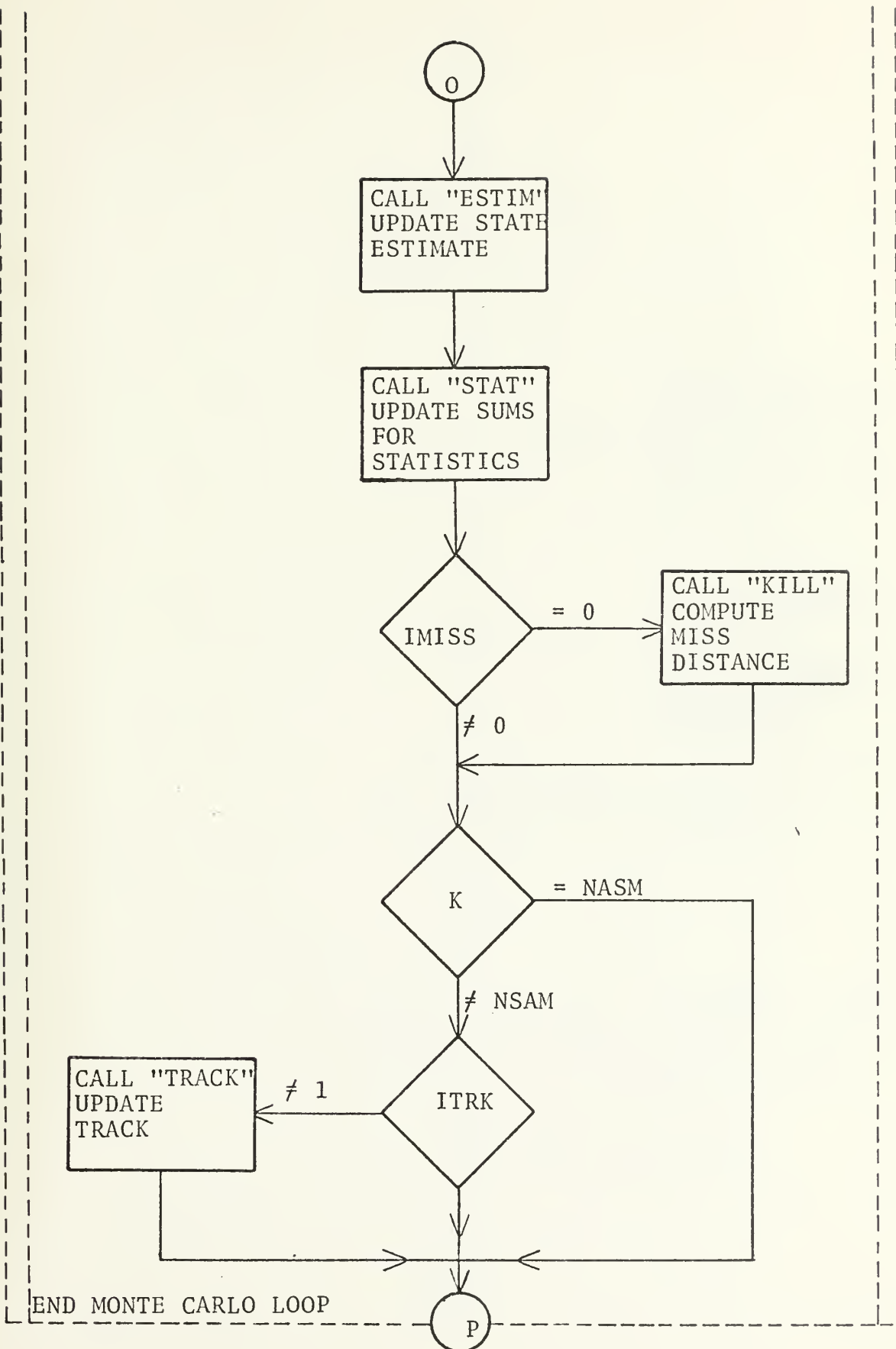


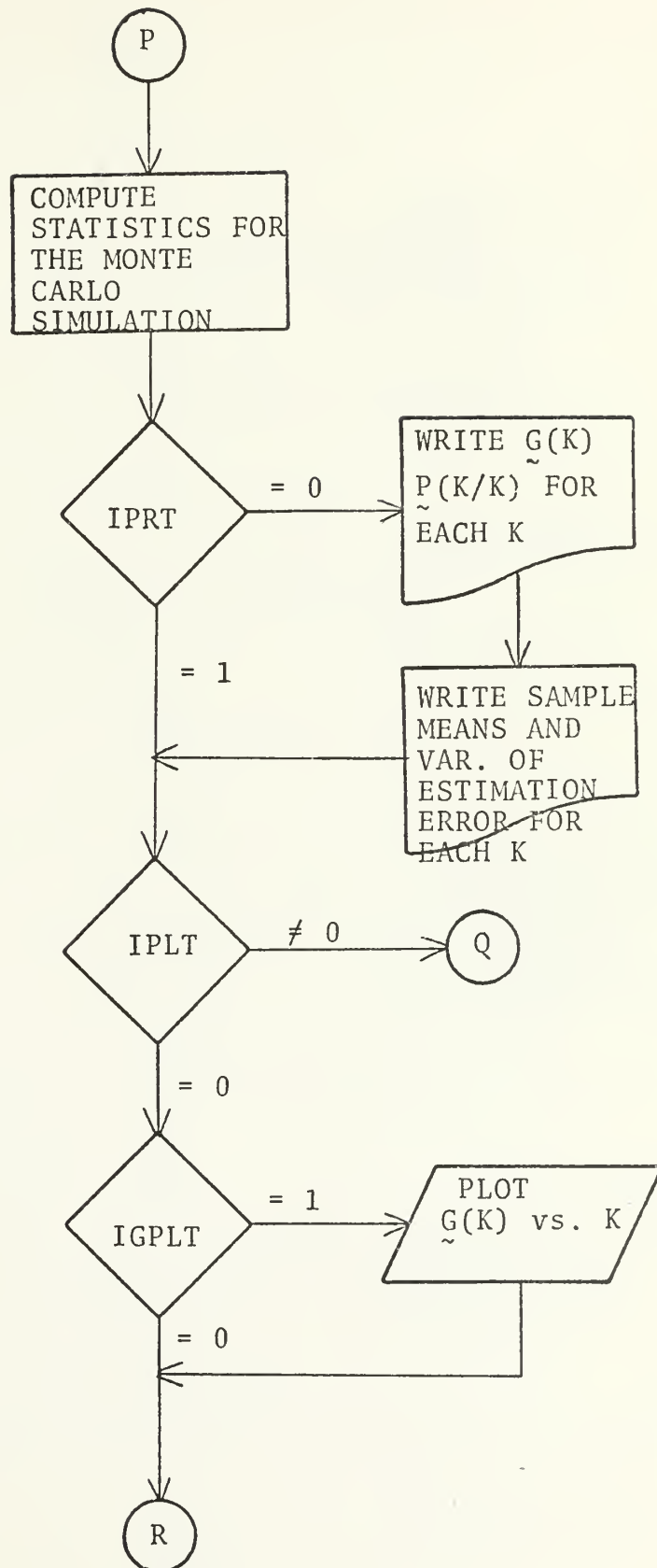


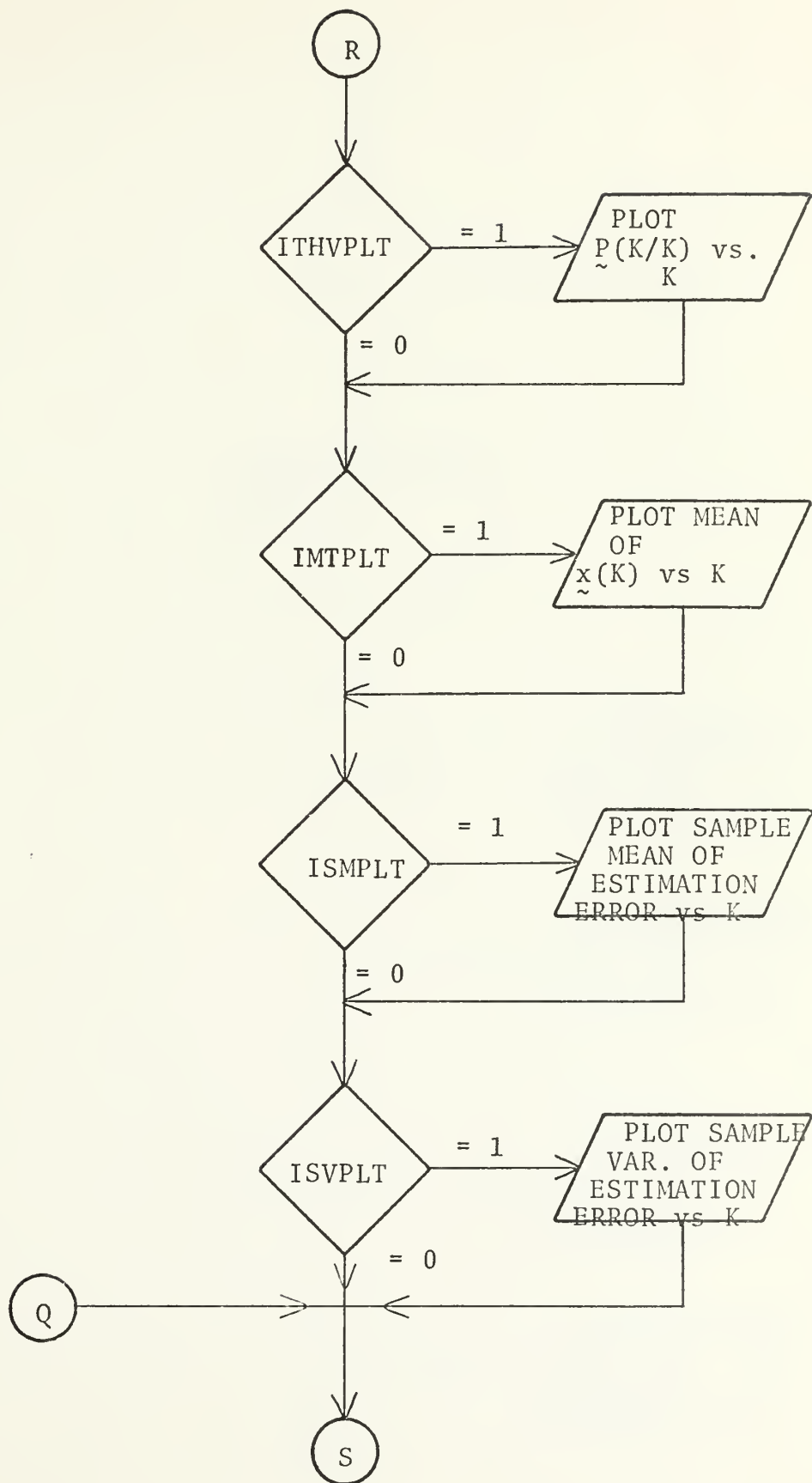


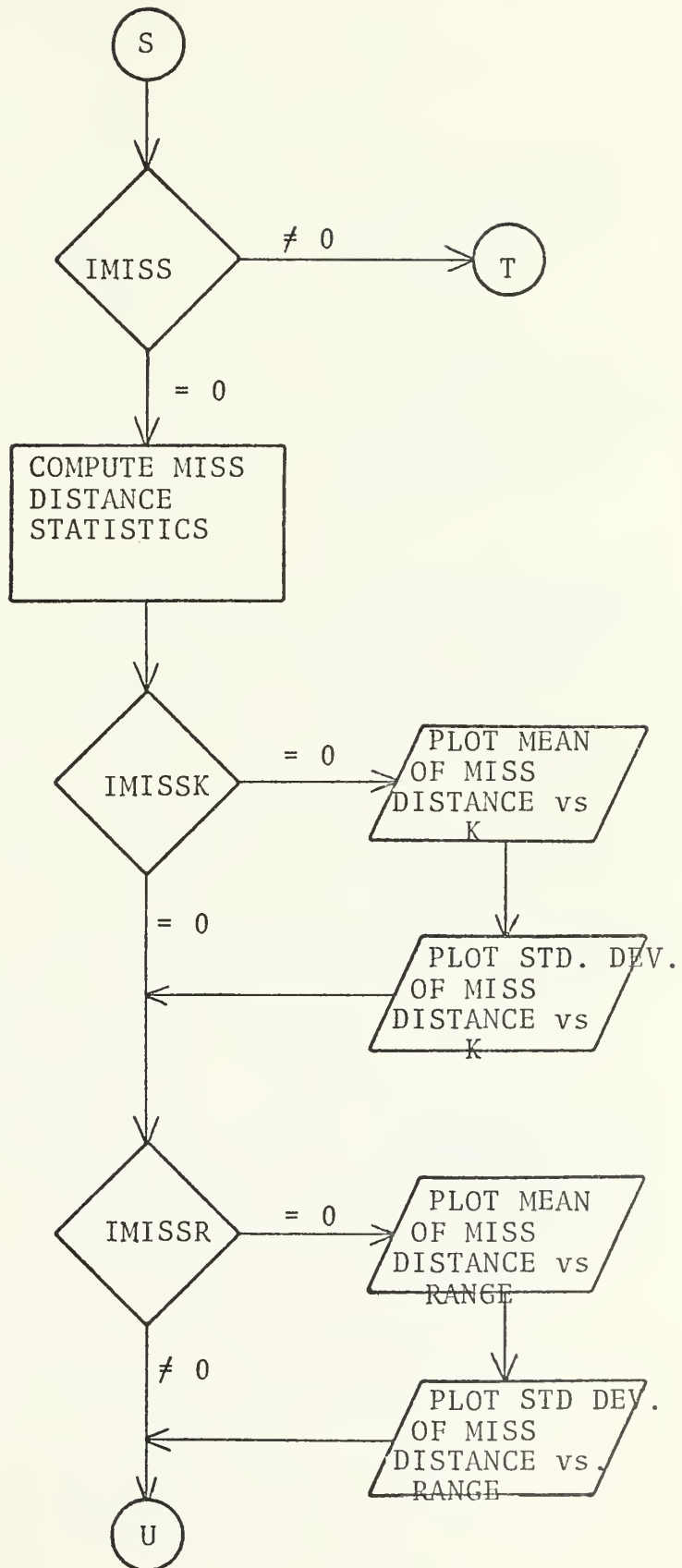


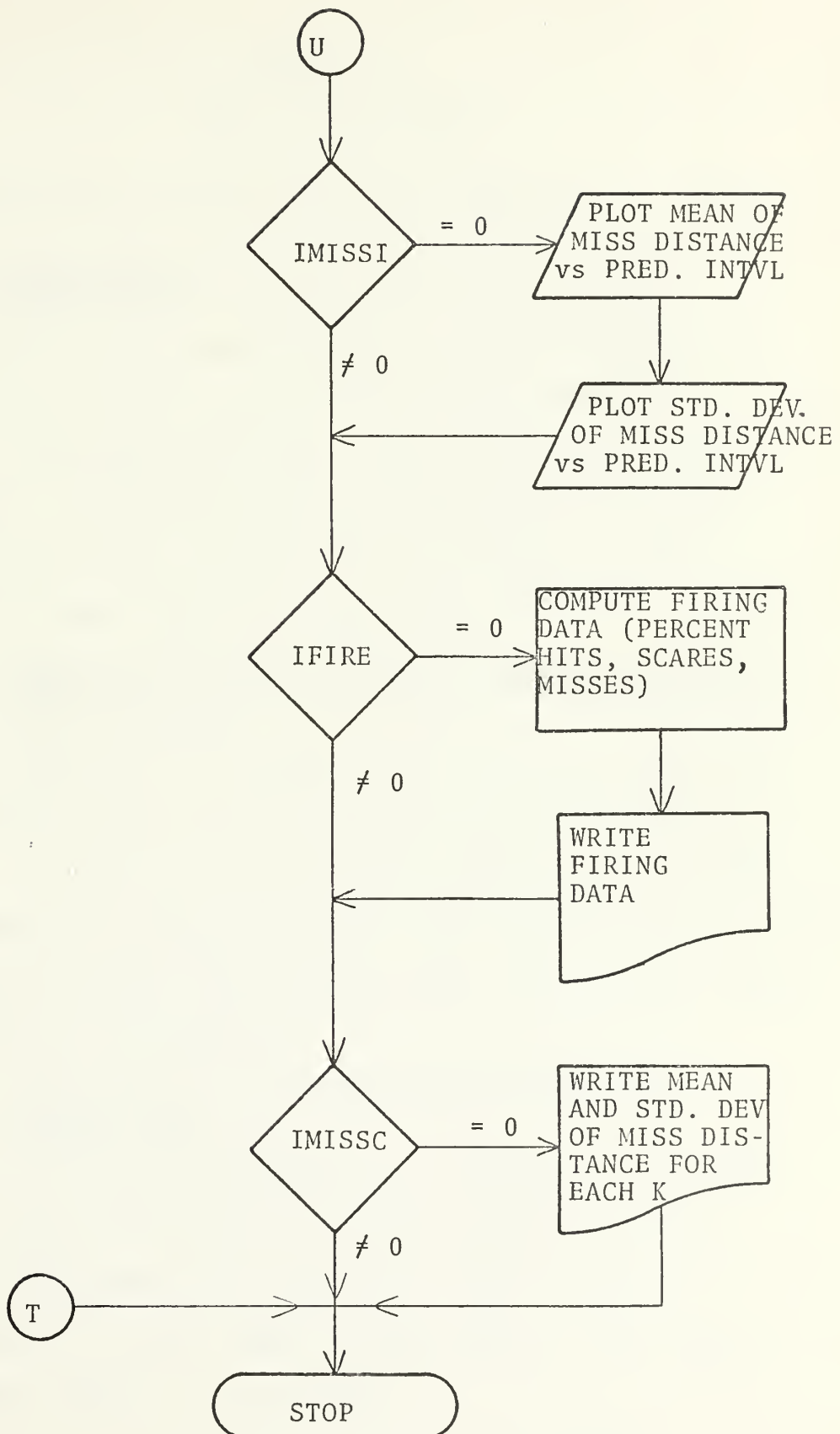












APPENDIX B
TEST TRAJECTORIES

B.1 DESCRIPTION OF TEST TRAJECTORY PROFILES

a. Trajectory 1; short-range, low speed, non-maneuvering

Trajectory 1 commences at a range of 3850 yards in a 30 degree dive with an initial velocity of -230 yds/sec (the negative velocity indicates motion toward the origin). The trajectory is terminated at a final range of 457 yards. No maneuver is executed. Sixty data points were taken at a 4-Hz rate.

b. Trajectory 2; short-range, low speed, moderate level maneuver

Trajectory 2 commences at a range of 3850 yards in a 30 degree dive with an initial velocity of -230 yds/sec. At 3450 yards the target executes a 2.0 g sin^2 planar maneuver lasting 13.5 sec. The maneuver is completed at the final sample range of 465 yards. Sixty data points were taken at a 4-Hz rate.

c. Trajectory 3; long-range, low speed, non-maneuvering

Trajectory 3 commences at a range of 9225 yards in a 30 degree dive with an initial velocity of -230 yds/sec. No maneuver is executed. The trajectory is terminated at a final range of 5832 yards. Sixty data points were taken at a 4-Hz rate.

d. Trajectory 4; long-range, low speed, moderate level maneuver

Trajectory 4 commences at a range of 9225 yards in a 30 degree dive with an initial velocity of -230 yds/sec. At 8822 yards, the target executes a 2.0 g sin^2 planar maneuver lasting 13.5 sec. The maneuver is completed at the final sample range of 5840 yards. Sixty data points were taken at a 4-Hz rate.

e. Trajectory 5; medium-range, high speed, evasive level maneuver

Trajectory 5 commences at a range of 4500 yards in a 20 degree dive with an initial velocity of -300 yds/sec. At 3900 yards the target executes a 6.0 g sin^2 planar maneuver lasting 10 sec. The maneuver is completed at a range of 914 yards and the trajectory is terminated at a final range of 90 yards. Sixty data points were taken at a 4-Hz rate.

f. Trajectory 6; long-range, high speed, evasive level maneuver

Trajectory 6 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -300 yds/sec. At 7500 yards the target executes a 6.0 g sin^2 planar maneuver lasting 10 sec. The maneuver is completed at a range of 4340 and the trajectory is terminated at a range of 3590 yards. Sixty data points were taken at a 4-Hz rate.

g. Trajectory 7; medium-range, high speed, non-maneuvering

Trajectory 7 commences at a range of 5000 yards in a 20 degree dive with an initial velocity of -300 yds/sec. No maneuver is executed. The trajectory is terminated at a final range of 575 yards. Sixty data points were taken at a 4-Hz rate.

h. Trajectory 8; long-range, high speed, non-maneuvering

Trajectory 8 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -300 yds/sec. No maneuver is executed. The trajectory is terminated at a final range of 3575 yards. Sixty data points were taken at a 4-Hz rate.

i. Trajectory 9; long-range, low speed, non-maneuvering

Trajectory 9 commences at a range of 8000 yards in 20 degree dive with an initial velocity of -200 yds/sec. No maneuver is executed. The trajectory is terminated at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

j. Trajectory 10; long-range, low speed, moderate level maneuver

Trajectory 10 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 4000 yards the target executes a 2.0 g sin^2 planar maneuver lasting 13.5 sec. The maneuver is completed at a range of 1200 yards and the trajectory is terminated at

the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

k. Trajectory 11; long-range, low speed, moderate level maneuver

Trajectory 11 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 2700 yards the target executes a 2.0 g sin^2 planar maneuver lasting 13.5 sec. The maneuver is completed at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

l. Trajectory 12; long-range, low speed, moderate level maneuver

Trajectory 12 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 3333 yards the target executes a 2.0 g sin^2 planar maneuver lasting 13.5 sec. The maneuver is completed at a range of 500 yards and the trajectory is terminated at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

m. Trajectory 13; long-range, low speed, moderate level maneuver

Trajectory 13 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 2700 yards the target executes a 2.5 g sin^2 planar maneuver lasting 10 sec. The maneuver is completed at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

n. Trajectory 14; long-range, low speed, moderate level maneuver

Trajectory 14 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 4000 yards the target executes a $2.5 g \sin^2$ planar maneuver lasting 10 sec. The maneuver is completed at a range of 600 yards and the trajectory is terminated at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

B.2 TRAJECTORY GENERATING PROGRAM

The trajectories of Section B.1 are the products of a tracking generating program written by Parr. A complete description of its use, development of the planar \sin^2 maneuver, program listing and several examples may be found in Parr's work [Ref. 4].

APPENDIX C

THE N-STEP PREDICTION PROBLEM

The ultimate test of any fire control system is its ability to destroy an engaged target. To this end, the determining factor is the accuracy with which it predicts the future position of the target, since launcher laying orders are derived from knowledge of the target's present and future states. This of course, neglects the response characteristics of the launcher and the weapon (projectile) ballistics, both of which are assumed here to have negligible effects on the end result.

The interval over which the predictions are made is derived from a straight line approximation of the plot of projectile Time of Flight (TOF) vs. the target's present slant range, for a 5"/54 CAL. projectile. For the interval in question, {TOF: TOF \in (0.0,10.0)} the straight line approximation takes on the value

$$\text{TOF} = 1.43 \times 10^{-3} \text{ (sec/yard) } \times \text{ (slant range) } \quad (\text{C.1})$$

The TOF interval, (0.0,10.0) was based on the assumption that the maximum effective range of the 5"/54 CAL gun against an air target is 7000 yds, corresponding to a TOF of approximately 10.0 seconds.

In the absence of any deterministic inputs to the system, the target's future position predicted over N samples is given by

$$\hat{\tilde{x}}(K+N/K) = \phi^N \hat{\tilde{x}}(K/K) \quad (C.2)$$

Raising the state transition matrix $\phi(K, K+1)$ to a decimal power presents some computational difficulties, so N is formed by computing the TOF from $\hat{x}_1(K/K)$ (the current estimate of the target's slant range) then truncating this value to give the integer N. For large N the difficulty of raising a (9 x 9) matrix to the Nth power still exists. Note, however, that the assumption of linear plant dynamics decouples the filter in each of three coordinate directions. Partitioning and decoupling we have for the system,

$$\tilde{x}(K+1) = \begin{bmatrix} \phi_{11} & 0 & 0 \\ 0 & \phi_{22} & 0 \\ 0 & 0 & \phi_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1(K) \\ \tilde{x}_2(K) \\ \tilde{x}_3(K) \end{bmatrix} + \begin{bmatrix} \Gamma_1 & 0 & 0 \\ 0 & \Gamma_2 & 0 \\ 0 & 0 & \Gamma_3 \end{bmatrix} \begin{bmatrix} \tilde{w}_1(K) \\ \tilde{w}_w(K) \\ \tilde{w}_3(K) \end{bmatrix} \quad (C.3)$$

where

$$\begin{aligned} \tilde{x}_1(K) &= [r, \dot{r}, \ddot{r}]^T & \tilde{w}_1(K) &= [w_1(K)] \\ \tilde{x}_2(K) &= [\phi, \dot{\phi}, \ddot{\phi}]^T & \tilde{w}_2(K) &= [w_2(K)] \\ \tilde{x}_3(K) &= [\theta, \dot{\theta}, \ddot{\theta}]^T & \tilde{w}_3(K) &= [w_3(K)] \end{aligned} \quad (C.4)$$

Further, note that for each of the models discussed in this thesis, each of the ϕ_{ii} $i = 1, 2, 3$ are lower triangular matrices. Following the method given by Parr [Ref. 4], the elements of the ϕ_{ii} $i = 1, 2, 3$ are operated on individually and the resulting ϕ_{ii}^N $i = 1, 2, 3$ are adjoined to form the system state transition matrix $\phi^N(K+N, K)$. The intermediate result follows quickly and is shown on the following page.

$$\begin{aligned}
 \phi_{ii}^N = & \left[\begin{array}{ccc} \phi_{ii}(1,1) & (\phi_{ii}(1,2))^N & \left\{ \phi_{ii}(1,3) \sum_{M=1}^N \phi_{ii}(3,3)^{M-1} \right. \\ & + \phi_{ii}(1,)\phi_{ii}(2,3) \sum_{M=1}^{N-1} (N-M)\phi_{ii}(3,3)^{M-1} \Big\} \\ 0 & \phi_{ii}(2,2) & \phi_{ii}(2,3) \sum_{M=1}^N \phi_{ii}(3,3)^{M-1} \\ 0 & 0 & \phi_{ii}(3,3)^N \end{array} \right] \quad (C.5)
 \end{aligned}$$

Finally, adjoining the ϕ_{ii}^N 's gives

$$\phi_{\sim}^N(K+N, K) = \begin{bmatrix} \phi_{11}^N & 0 & 0 \\ 0 & \phi_{22}^N & 0 \\ 0 & 0 & \phi_{33}^N \end{bmatrix} \quad (C.6)$$

The result shown in Equation (C.5) is easily programmed and involves simple manipulations without the need for extensive matrix multiplication. Note that the result shown in Equation (C.5) is equivalent to (for the single dimension)

$$r(K+N) = r(K) + \dot{r}(K)[(N-K)T] + \frac{1}{2} \ddot{r}(K)[(N-K)T]^2 \quad (C.7)$$

For the case of the constant-velocity model, $\phi_{ii}(1,3)$, $\phi_{ii}(2,3)$, and $\phi_{ii}(3,j)$ $j = 1,2,3$ are set to zero. The prediction is then easily found by using Equation (C.2).

The magnitude of the N-step prediction error is defined to be the magnitude of a position vector, say $\overrightarrow{\text{ERRNET}}$, drawn between the estimate at time $t = (K+N)T$, $\hat{x}(K+N/K)$, and the reference trajectory, $x(K+N)$

$$||\overrightarrow{\text{ERRNET}}|| = ||x(K+N) - \hat{x}(K+N/K)|| \quad (C.8)$$

Note that no reference is made to the direction of the error vector since the previously defined kill region is a sphere of constant radius centered at the reference position $x(K+N)$.

The primary interest lies in the following three regions

- 1) $||\overrightarrow{\text{ERRNET}}|| \leq 8.0 \text{ yds}$ hit
- 2) $8.0 < ||\overrightarrow{\text{ERRNET}}|| \leq 16.0 \text{ yds.}$ scare
- 3) $||\overrightarrow{\text{ERRNET}}|| > 16.0 \text{ yds}$ miss

To avoid the necessity for coordinate conversion, the error vector is determined by a simple rotation of the polar plane followed by applications of the law of cosines. Consider the situation depicted in Figure 4, where points a and b represent $\hat{x}(K+N)$ and $\hat{x}(K+N/K)$ (position only). Then from Equation (C.8)

$$||\overrightarrow{\text{ERRNET}}|| = ||\overrightarrow{ab}|| = ||\overrightarrow{ba}|| \quad (\text{C.9})$$

If the polar plane in Figure 4 is rotated through some angle β about an axis constructed by drawing a line through the origin connecting the points (090,270) (keeping r_1 and r_2 fixed) Figure 5 is obtained. The angle of rotation is defined as

$$\beta = \min(\theta_1, \theta_2) \quad (\text{C.10})$$

Point now lies on the rotated polar plane, the position of point b remains fixed, and point c represents the fixed range projection of point b onto the polar plane. Some rather obvious geometric relationships follow immediately;

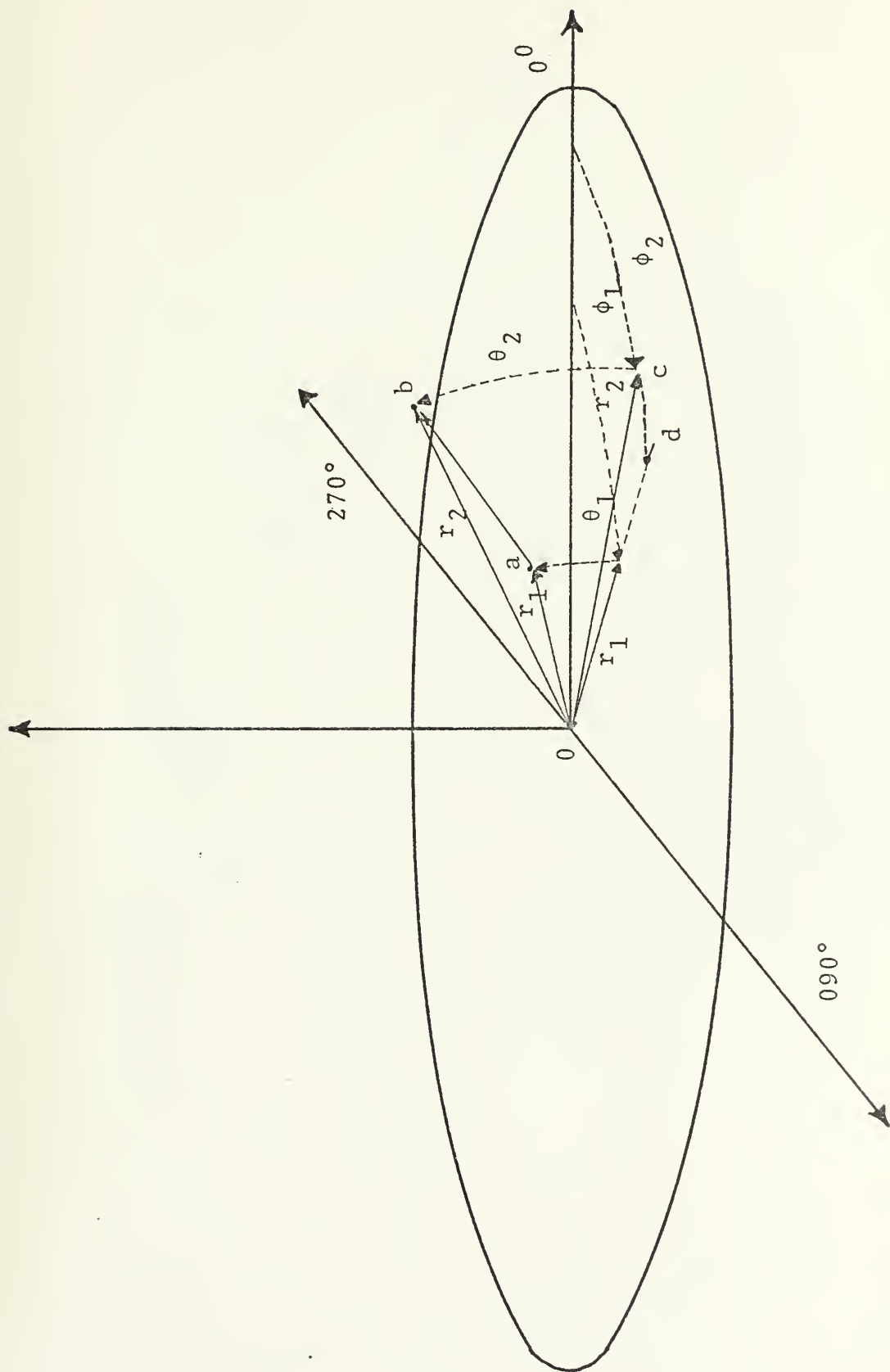


Fig. 4. GEOMETRY OF THE PREDICTED POSITION ERROR PROBLEM
IN A SPHERICAL COORDINATE SYSTEM

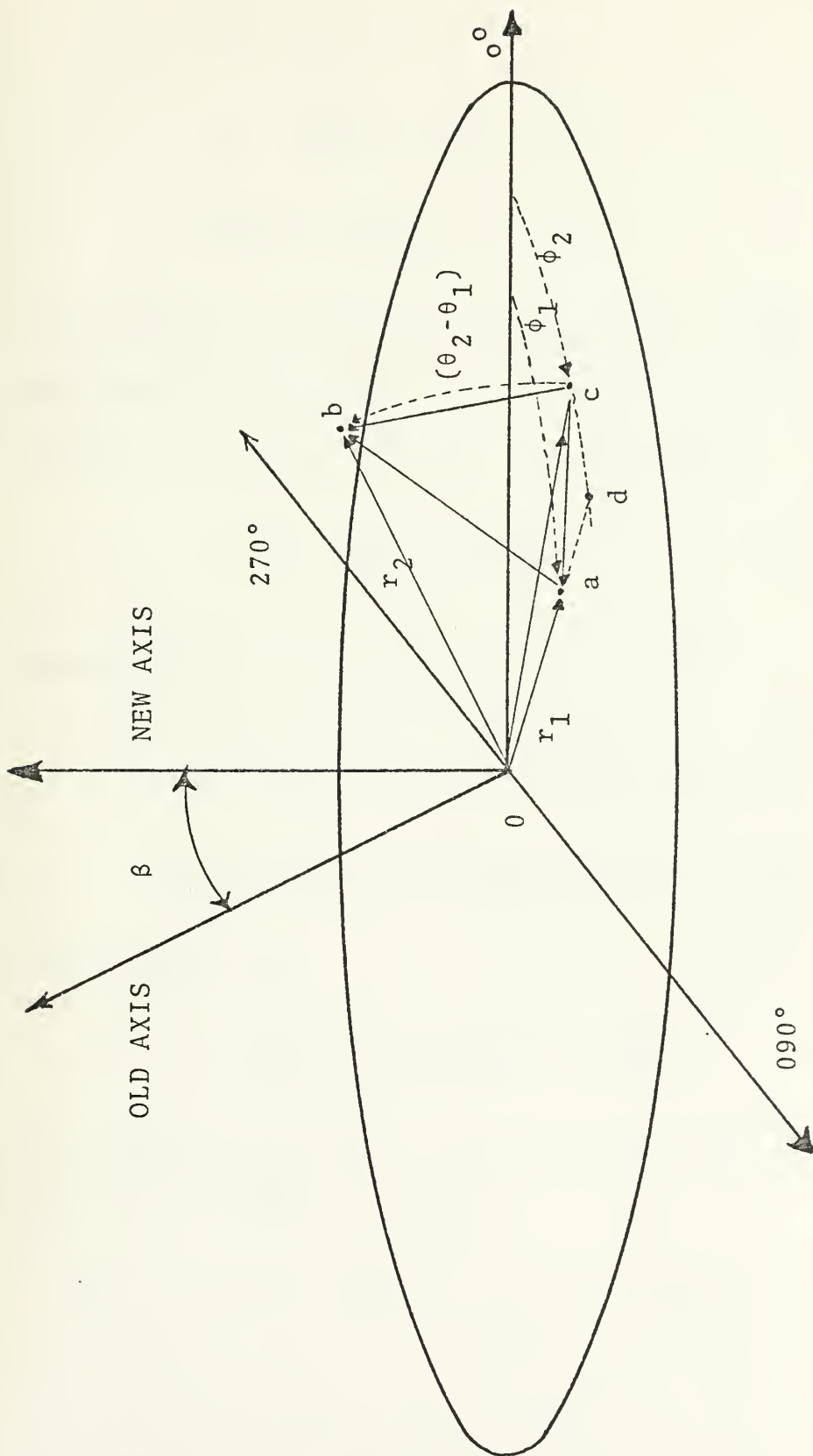


Fig. 5. THE POLAR PLANE AFTER ROTATION

$$\Delta\theta = \angle boc = |\theta_1 - \theta_2|$$

$$\Delta R = |\vec{ad}| = |r_1 - r_2| \quad (C.11)$$

$$|\vec{od}| = \max(r_1, r_2)$$

The problem then remains of finding $|ab|$. Figure 6 shows the projection onto the rotated polar plane. Note that triangle cod is an isosceles triangle, then

$$\angle cdo = \frac{180 - \Delta\phi}{2} \quad (C.12)$$

Applying the law of cosines $|\vec{ac}|$ is given by

$$|\vec{ac}| = \{(|\vec{ad}|)^2 + [(2.0)(\sin\frac{\Delta\phi}{2.0})(|\vec{od}|)]^2 - (2.0)(|\vec{ad}|)(2.0)(\sin\frac{\Delta\phi}{2.0})(|\vec{od}|) \cos\angle cdo\}^{1/2} \quad (C.13)$$

$|\vec{bc}|$ is given by

$$|\vec{bc}| = (2.0)(|\vec{od}|)(\sin\frac{\Delta\theta}{2.0}) \quad (C.14)$$

Again applying the law of cosines and noting that

$\angle bcd = \angle bco$, $|\vec{ab}|$ is given by

$$|\vec{ab}| = \{|\vec{bc}|^2 + |\vec{ac}|^2 - (2.0)(|\vec{bc}|)(|\vec{ac}|)\cos(\angle bcd)\}^{1/2} \quad (C.15)$$

Thus, the magnitude of the N-step prediction error has been determined without the need for any coordinate conversion. The validity and accuracy of this means of determining the

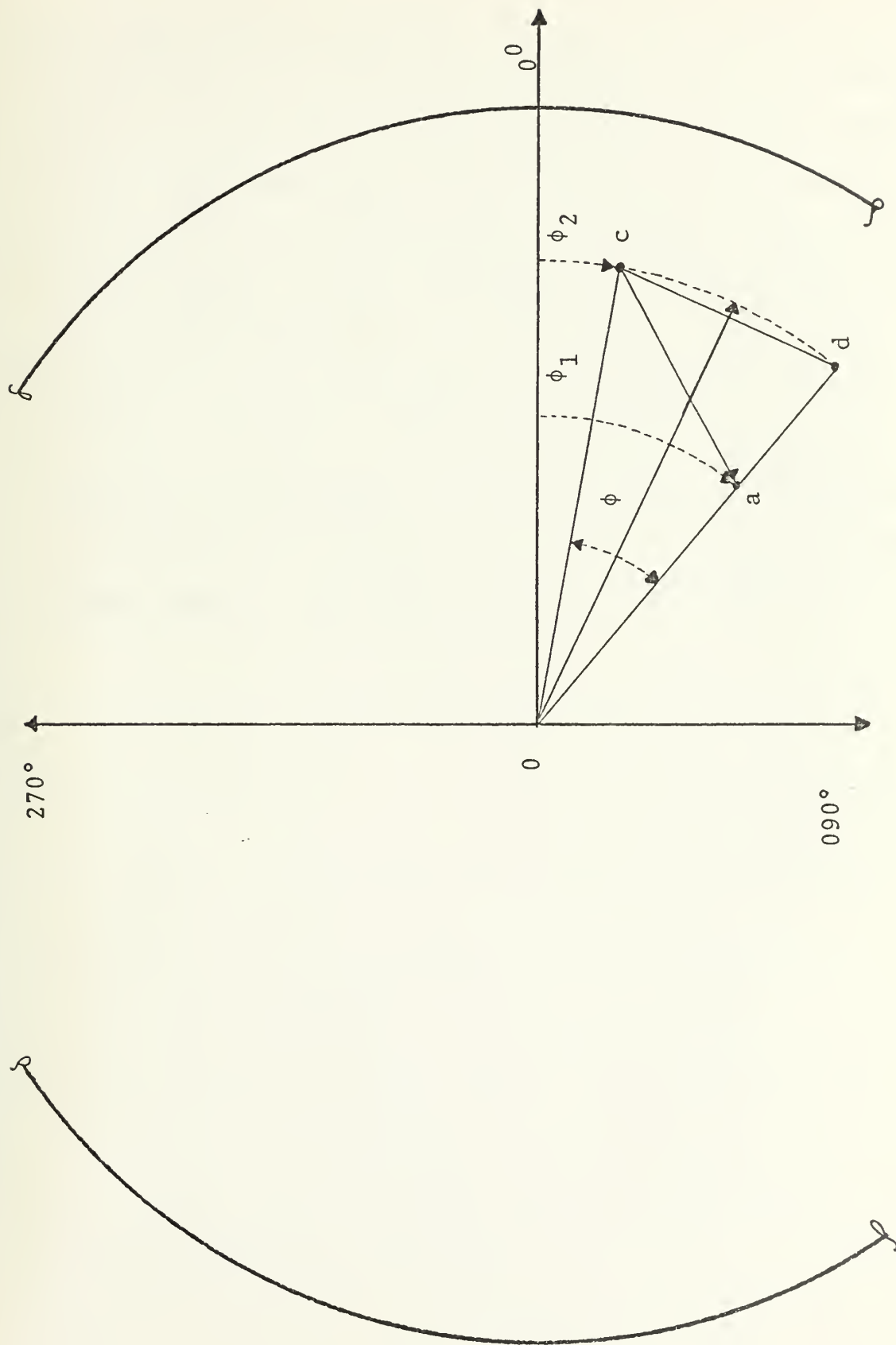


Fig. 6. FIXED RANGE PROJECTION ONTO THE ROTATED POLAR PLANE

error magnitude was verified by conversion of $\tilde{x}(K+N)$ and $\hat{\tilde{x}}(K+N/K)$ to the Cartesian Coordinate system and determining the magnitude of the error vector by standard means. In every case investigated, the results given by the two different methods were in complete agreement. Again note that no attempt is made to determine the direction of the error vector $|\overrightarrow{ab}|$ since the regions of interest specify concentric spheres of constant radii centered at $\tilde{x}(K+N)$. The directionality of the error vector is eliminated in SUBROUTINE KILL through the use of the absolute values of the differences in bearing and elevation, and by using the maximum value of the actual and estimated ranges.

BIBLIOGRAPHY

1. NAVAL POSTGRADUATE SCHOOL Report NPS-52DE8041A, Adaptive Tracking of Maneuvering Targets, by J. S. Demetry and H. A. Titus, 15 April 1968.
2. NAVAL POSTGRADUATE SCHOOL Report NPS-52KI74101, Evaluation of State Estimators and Predictors for Fire Control Systems, by D. E. Kirk, October 1974.
3. Ketron, M. G., Monte Carlo Evaluation of Digital Filters for Fire Control Systems, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, June 1974.
4. Parr, J. M., An Estimator for an Anti-Aircraft Gun Fire Control System, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, December 1974.
5. McKinley, D. H., Modeling Maneuvering Airborne Target with a Coloring Noise Filter, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, June 1974.
6. Iida, T., Monte Carlo Evaluation of Digital Filters for Fire Control Systems, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, December 1975.
7. Gelb, A., ed., Applied Optimal Estimation, MIT Press, 1974.
8. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, 1970.
9. Sorenson, H. W., "Kalman Filtering Techniques," Advances in Control Systems, C.T. Leonedes, ed., Academic Press, vol. 3, 1966.
10. Singer, R. A., "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, vol. 6, July 1970.
11. THE ANALYTICAL SCIENCES CORPORATION Report TR-387-1, Adaptive Tracking Filter Design and Evaluation for Gun Fire Control Systems, by C. F. Price and C. M. Brown, 23 January 1974.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4. Professor D. E. Kirk, Code 52 Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	3
5. LT Robert K. Brands 2173 Stanford Avenue St. Paul, Minnesota 55105	2
6. Mr. H. A. Parish System Integration and Engineering Division Code 4320 Naval Ship Weapon Systems Engineering Station Port Hueneme, California 93043	1
7. Dr. B. L. Clark System Integration Division Naval Surface Weapons Center Dahlgren, Virginia	1
8. Mr. C. F. Price The Analytical Sciences Corporation 6 Jacob Way Reading, Massachusetts 01867	1
9. Professor H. A. Titus, Code 52 Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1

9 JUN 76
31 AUG 80

23646
25787

Thesis
B7988
c.1

Brands

An alternative to
nonlinear estimation
in gun fire control
systems.

9 JUN 76
31 AUG 80

23646
25787

Thesis
B7988
c.1

Brands

An alternative to
nonlinear estimation
in gun fire control
systems.

104087

thesB7988

An alternative to nonlinear estimation i



3 2768 001 01624 9

DUDLEY KNOX LIBRARY